Universidad Carlos III de Madrid

**Ingeniería de Telecomunicación**

# PROYECTO FIN DE CARRERA

# Desarrollo de un asistente multimodal para la gestión de partidos de balonmano mediante dispositivos móviles

**Autor:** Jonathan Caballero Álvarez

**Tutor / Director:** Dr. David Griol Barres

**Junio de 2015**

EL TRIBUNAL

Presidente: _____

Vocal:

_____

Secretario:

_____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

# Resumen

El objetivo principal de este Proyecto Fin de Carrera es el desarrollo de una aplicación que se pueda utilizar a través de un sistema de diálogo y que se ejecute en dispositivos móviles que utilizan el sistema operativo Android, debido a que es el S.O. más extendido actualmente en el mercado.

Se ha decidido centrar la aplicación en un mundo que demanda cada vez más este tipo de tecnología por el gran volumen de datos que se necesitan manejar y muchas veces en tiempo real, debido a que es necesaria la toma de decisiones rápidas basadas en la centralización de esta información. Se trata del mundo del deporte, la aplicación se centra en un deporte en concreto, el balonmano, aunque no sería difícilmente adaptable a otros deportes de equipo.

La aplicación desarrollada permite realizar las estadísticas de un partido de balonmano en tiempo real utilizando como medio de comunicación con el dispositivo móvil además del tradicional medio táctil como medio de entrada de datos el medio vocal. Además, la aplicación permite al usuario almacenar el resultado de estas estadísticas para poder consultarlas post-partido, así como almacenar los resultados derivados de las mismas de forma automática.

En la actualidad la amplia gama existente de aplicaciones móviles garantiza la existencia de muchas aplicaciones disponibles a tal fin de realizar estadísticas en tiempo real para deportes. Hemos querido llevarlo un poco más allá incluyendo la posibilidad de la interacción oral para hacer más interesante al usuario esta aplicación y que pueda resultar su diferenciación de las demás.

**Palabras clave:** Sistemas de diálogo, interacción oral, deporte, balonmano, sistema de voz, dispositivos móviles, Android.

# Abstract

The main objective of this Bachelor Project is to develop an app that can be used through a dialogue system and running on mobile devices using Android, because it is O.S. currently most extended in the market.

I decided to focus my app in a world that increasingly demands this type of technology by the large volume of data required to manage in real time, because it is necessary to take quick decisions based on this information. This is the sport world, my app focuses on a particular sport, handball, although it would not be hard adaptable to other team sports.

The app I made allows make the statistics of a handball match in real time using as a means of communication with the mobile device a voice system, besides the traditional touch medium. In addition, app allows the user to store the result of these statistics for reference by post-match as well as store the results derived from them automatically.

At present the wide range of mobile applications ensures the existence of many apps available for the purpose of performing real-time statistics for sports. I wanted to take it a further including the possibility of oral interaction to make it interesting to the user this app and that may be their differentiation from other.

**Keywords:** Dialogue systems, oral interaction, sport, handball, voice systems, mobile devices, Android.

# Índice general

# Índice de Figuras

# Índice de Tablas

# Estructura del documento

En este apartado se pretende dar una breve descripción del contenido de cada uno de los capítulos que componen esta memoria. La finalidad es que el lector de esta memoria tenga una idea general de cada uno de ellos con anterioridad y facilite la lectura de los mismos.

**Capítulo 1: Introducción y objetivos.** En este capítulo se pretende dar una idea general de la motivación que ha llevado a cabo la realización de este Proyecto Fin de Carrera. El marco cultural de las nuevas tecnologías en que está envuelto y la posibilidad de cubrir una necesidad real con el resultado de la aplicación.

**Capítulo 2: Estado del Arte.** Este capítulo trata de situar al lector en el contexto actual. Describe de forma breve la situación actual de los sistemas de Diálogo así como de las tecnologías/sistemas necesarios para la implementación de este proyecto.

**Capítulo 3: Descripción de la aplicación y de los módulos del sistema.** Este capítulo describe de forma detallada cada uno de los módulos en que está dividido el sistema, describiendo su funcionalidad, arquitectura, comandos de voz, diagrama de flujo y escenarios de uso.

**Capítulo 4: Pruebas del Sistema.** En este capítulo se hará una breve descripción de las pruebas a realizar para la validación del sistema.

**Capítulo 5: Evaluación del Sistema.** En este capítulo se lleva a cabo una evaluación del sistema y una serie de valoraciones subjetivas de potenciales usuarios de la aplicación realizada.

**Capítulo 6: Presupuesto.** Este apartado contiene un análisis de los costes del diseño y desarrollo del proyecto, detallando el coste de material, personal y resto de recursos necesarios para llevar a cabo su realización.

**Capítulo 7: Conclusiones y trabajo futuro.** Se exponen las principales ideas, cuestiones, y conclusiones derivadas de la realización del proyecto, así como las posibles líneas de investigación a partir de este proyecto se podrían generar.

**Glosario.** En este apartado se recopilan los principales términos y conceptos técnicos utilizados en la memoria, con el objetivo de facilitar su comprensión al lector.

**Bibliografía.** En este apartado se reflejan las citas bibliográficas que se han consultado para la realización de este proyecto fin de carrera.

# Capítulo 1. Introducción y objetivos

## 1.1 Introducción

En los tiempos que corren la tecnología cada vez más se está introduciendo en la vida cotidiana del ser humano. Estamos rodeados de un sin fin de elementos tecnológicos que nos permiten que dicha vida cotidiana sea más fácil y nos permita realizar tareas de forma más rápida y eficiente.

Sobre todo en los últimos años uno de estos elementos tecnológicos que ocupan un gran número de horas en nuestras tareas cotidianas y que nos permiten realizar muchas tareas que hace no mucho tiempo eran impensables son los dispositivos móviles, más en concreto por estar su utilización más extendida el caso de los Smartphone y las tablets.

Por dar unos datos del impacto de estos dispositivos en el ámbito nacional, un estudio de IAB Spain a principios de 2014 sobre la utilización en el año anterior por los españoles recoge los siguientes datos que son muestra de que dichos dispositivos forman parte de una parte importante de nuestro día a día **[IAS12]** (Figuras 1 y 2):

- Los Smartphone pasan de cuota de mercado del 58% al 80%.
- Las tablets se consolidan con un crecimiento del 23% hasta el 43% en un solo año.
- Los dispositivos móviles igualan a los portátiles en cuanto al acceso diario a Internet, el 88% acceden desde el portátil, mientras que el 86% lo hace desde su Smartphone.
- El acceso a internet mediante tablets crece hasta el 45%.

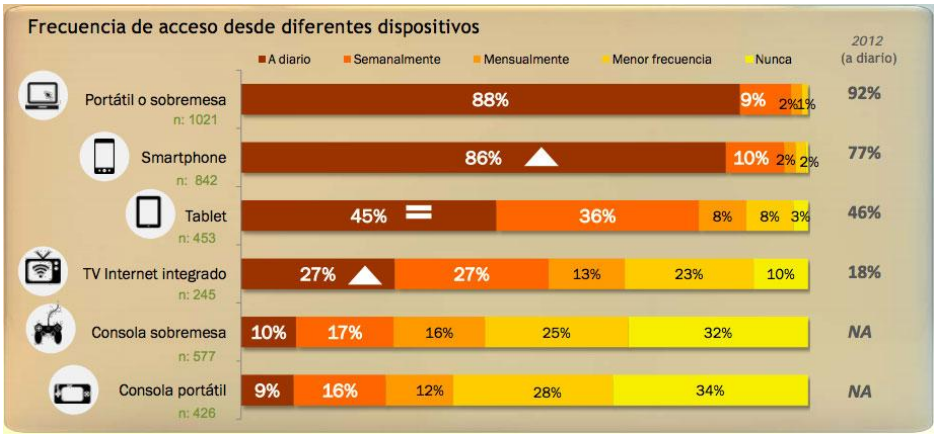



Figura 1. Frecuencia de acceso a Internet desde diferentes dispositivos [IAS12]

- El uso de acceso a Internet mediante dispositivos móviles es de 2 horas y 30 minutos de media. 8 de cada 10 usuarios de Smartphone invierte más de una hora en la Red.
- La media del uso de tablets se coloca en 2 horas y 6 minutos de media.
- Un 78% de los españoles consulta su correo electrónico desde su dispositivo móvil, al menos una vez a la semana. Creciendo un 19% en el último año.
- La utilización de los Smartphone por franja de edad y la comparativa respecto al año anterior se puede ver en la Figura 2.

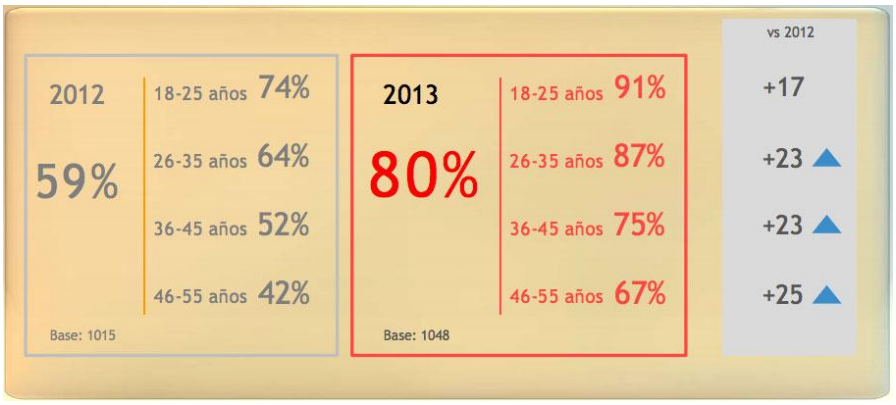



Figura 2. Utilización de Smarhtpone por franjas de edad [IAS12]

Este hecho nos lleva a la situación de que hay que dotarnos de elementos de comunicación que nos permitan la interacción hombre-máquina para poder utilizar dichos elementos tecnológicos.

# 1.2 Objetivos

El objetivo de este Proyecto Fin de Carrera es el desarrollo de una aplicación que se pueda utilizar en dispositivos móviles, y que sea capaz de

funcionar en el Sistema Operativo Móvil llamado Android, diseñado para este tipo de dispositivos.

Dicha aplicación estará basada en una comunicación hombre-máquina que está cada vez más extendida entre los dispositivos móviles, como son los comandos de voz, a través de los cuales podemos interactuar con el dispositivo sin necesidad de tocarlo y sin utilizar el método tradicional de los comandos táctiles.

Esta interacción hombre máquina que desarrollará la aplicación por tanto se debe dividir en dos fases (Figura 3):

- **Interacción Usuario – Aplicación:** Se trata de lo que se conoce en inglés como *Voice Recognition*. Es la parte de la comunicación en que el usuario a través del método más fluido de comunicación que tiene el ser humano, como es la voz, comunica a la aplicación las acciones que desea realizar. La aplicación debe ser capaz de interpretar los datos aportados por el usuario y actuar en consecuencia.
- **Interacción Aplicación – Usuario:** Esta parte se conoce en inglés como *Text to Speech.* En esta parte de la comunicación la aplicación tiene una respuesta que dar al usuario, normalmente esto se describe como una frase en texto que tiene que convertir a comandos de voz, de ahí el nombre en inglés.


Figura 3. Diagrama de ejecución de la aplicación

# 1.3 Motivación

En mi caso la motivación de este Proyecto Fin de Carrera es poder unificar el objetivo del mismo con la pasión que tengo hacia un deporte en particular, como es el balonmano.

Tradicionalmente en el mundo del deporte en general, y en el del balonmano de forma más concreto en el caso que me ocupa, la competitividad es el elemento clave que lleva a los equipos a la búsqueda de resultados positivos para lograr éxitos. En esta búsqueda de resultados se hace patente la necesidad de estudiar a tu propio equipo y sobre todo a los rivales a los que te enfrentas con el objetivo de buscar debilidades propias y ajenas, así como puntos fuertes permitiendo utilizarlos para beneficio propio. Y en este sentido se trata de

recoger en tiempo real información de los encuentros, entre otras cosas, las estadísticas de los mismos, de forma que los miembros del cuerpo técnico correspondiente puedan analizarlas con el fin de buscarles un beneficio.

De esta forma, hemos identificado una necesidad concreta que puede ser cubierta cumpliendo el objetivo planteado en este Proyecto Fin de Carrera. Basándome en la experiencia personal que me da ser oficial de este deporte y más en concreto del puesto que se conoce como *Delegado de Equipo*, cuya función, entre otras es tomar datos en tiempo real de las acciones que transcurren durante un partido para poder ser analizadas durante el mismo o de forma posterior por el *Entrenador*.

La necesidad surge del dinamismo del balonmano, en el cual las acciones se suceden de forma muy rápida y se enlazan de forma muy consecutiva, de forma que en muchas ocasiones si no se está atento es fácil perderse alguna. Tradicionalmente para poder registrar estos datos se utilizaba bolígrafo y papel, esto conllevaba que en el momento de ir a anotar alguna acción es muy fácil perderse la consecutiva, o tener que memorizar varias que ocurren de forma consecutiva para anotarlas posteriormente todas juntas. En la actualidad es fácil encontrarse con la utilización de dispositivos móviles en los que se pueden descargar aplicaciones para poder registrar estas estadísticas, pero el problema en cuestión sigue existiendo, puesto que utilizando la entrada de datos habitual para estos dispositivos como es la táctil, mientras el usuario busca el botón/es adecuado/s para registrar la acción que acaba de suceder corre el riesgo de que ocurra alguna otra durante el partido que no pueda ver puesto que está ocupado mirando el dispositivo móvil.

De este problema, surge la necesidad de encontrar otro método de entrada de datos que permita al usuario de la aplicación la posibilidad de no apartar la vista del partido para poder ser capaz de visualizar todas las acciones que ocurren en el mismo. La solución planteada es utilizar el objetivo de este Proyecto Fin de Carrera para crear una aplicación para registrar acciones y estadísticas de un partido de balonmano cuya entrada de datos no sea la habitual de método táctil, sino que se utilicen comandos de voz con este fin.

# 1.4 Antecedentes

El mundo de las aplicaciones en Android es muy amplio, y es fácil encontrar mucha variedad de aplicaciones sobre un mismo tema. Y por tanto el mundo del deporte no es una excepción. No hay más que darse una vuelta por *Google Play* para encontrar varias aplicaciones que realizan funcionalidades muy similares, si no las mismas que la aplicación planteada en este Proyecto Fin de Carrera. A continuación se describen las más importantes:

## Estadísticas Balonmano en su versión Demo

- Aplicación gratuita para su versión para 1 partido y de pago en su versión completa.
- Es compleja de configurar para poder comenzar con la introducción de estadísticas en tiempo real de un partido, puesto que es necesario introducir mucha información antes de comenzar. Sin embargo es muy completa en la información solicitada: nombre, nacionalidad, fecha de nacimiento…., puede resultar muy útil para alguien que necesite gestionar toda esta información.
- Sencilla y fluida para la introducción de datos de las estadísticas. Pero únicamente permite modo de entrada de datos de forma manual a través de la pantalla táctil. En la Figura 4 se puede ver el logo de esta aplicación en *Google Play*

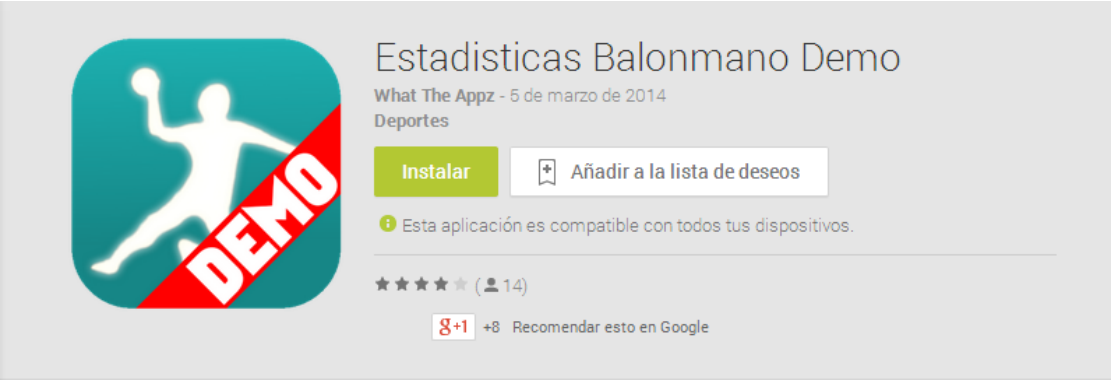



Figura 4. Aplicación *Estadísticas Balonmano* en su versión Demo

## Estadísticas Handball

- Aplicación gratuita.
- Aplicación muy básica y sencilla para guardar las estadísticas del equipo propio durante un partido. No permite introducción de las estadísticas del equipo contrario.
- Muy rápida y fácil para introducir las acciones pero muy limitadas. Únicamente permite modo de entrada de datos de forma manual a través de la pantalla táctil. En la Figura 5 se pude ver el logo de esta aplicación en *Google Play.*


Figura 5. Aplicación *Estadísticas Handball*

## Handball Scorekeeper Lite

- Aplicación gratuita para su versión para 1 partido y de pago en su versión completa.
- No tiene versión en castellano.
- Gestión rápida de los partidos para su inicio, únicamente es necesario introducir los nombres de los jugadores y equipos. Sin embargo para alguien que necesite más
- Aplicación muy sencilla en interfaz y que tiene acciones muy limitadas, sin apenas estadísticas. Permitiendo prácticamente de forma única anotar los goles y las amonestaciones de los jugadores. Puede ser insuficiente para alguien que necesite guardar más información de los jugadores. En la figura 6 se pude ver el logo de esta aplicación en *Google Play.*


Figura 6. Aplicación *Handball Scorekeeper Lite*

**Handball Analyse THSA-Extreme**

- Aplicación orientada a profesionales, de ahí que sea de pago y de ahí que tenga un alto precio.
- No está disponible en castellano.
- Es muy completa en todos los sentidos. Tiene una interfaz muy visual que permite introducir todo tipo de información y de estadísticas, así como recrear jugadas realizadas durante el partido de forma visual.
- Posee una completa y amplia Base de Datos para estadísticas.
- Compleja en su utilización, requiere mucho tiempo de aprendizaje para su uso. En la figura 7 se pude ver el logo de esta aplicación en *Google Play.*


Figura 7. Aplicación *Handball Analyse THSA-Extreme*

# 1.5 Fases del Desarrollo

El presente proyecto ha sido planificado en diferentes fases de desarrollo y se pueden agrupar en tres grandes grupos:

- **Fase de estudio**
  - **Sistemas de diálogo:** Adquisición de información acerca de que son, sus utilidades y cómo funcionan los sistemas de diálogo.
  - **Sistema operativo Android:** Adquirir toda la información posible sobre el estado actual del proyecto de google sobre su sistema operativo para dispositivos móviles Android.
    - Número de desarrollo actual: 4.4 KitKat.
    - IDE para el desarrollo: Eclipse.
    - Lenguaje de programación utilizado para desarrollo: Java.

- **Fase de planificación**
    - **Bases de Datos:** Planificación de la estructura de las bases de datos (BBDD) necesarias para guardar toda la información referente a las estadísticas que queremos guardar de los partidos.
    - **Interfaz gráfica:** Análisis y establecimiento de las pantallas necesarias para navegar por todas las funciones. Esto supone la división de las interfaces en diferentes líneas de flujo para cubrir todas las funcionalidades del sistema.
    - **Reconocimiento de voz:** Definición de las pautas necesarias para las acciones a realizar en función de las indicaciones del usuario.
    - **Comandos de voz:** Establecimiento de los comandos de voz necesarios para interactuar con la aplicación en cada una de las pantallas.
- **Fase de desarrollo**
    - **La interfaz gráfica:** Diseño de la parte visual de cada una de las interfaces, enlace entre las mismas para llevar a cabo un flujo natural de la aplicación y código necesario para aportar funcionalidad.
    - **Comunicación con BBDD:** Programación del código necesario para comunicar las interfaces de la aplicación con la BBDD ya creada para poder consultar, guardar y manipular datos.
    - **Análisis de Voz:** Desarrollo de las acciones que realiza la aplicación y el flujo seguido por las interfaces tras las indicaciones del usuario.
    - **Síntesis de Voz:** Desarrollo de las respuestas a dar por la aplicación en función del flujo desarrollado por el usuario al interactuar con las interfaces.
- **Fase de pruebas**
    - **Pruebas de flujo de la interfaz gráfica:** Pruebas para comprobar que la navegación entre las interfaces de la aplicación es correcta y funciona de forma adecuada.
    - **Pruebas funcionales de la aplicación:** Pruebas para comprobar que la aplicación cumple con las funcionalidades esperadas.
- **Fase de evaluación**
    - **Evaluación por parte de usuarios finales** de la aplicación desarrollada y recopilación de opiniones subjetivas de los mismos para comprobar el grado de satisfacción con la aplicación.

- **Evaluación estadística de la aplicación** mediante medidas del número de reconocimientos correctos, reconocimientos fallidos, tiempo de respuesta de la aplicación y de análisis de las frases del usuario.

- **<u>Fase de documentación</u>**
  - **Memoria del Proyecto Final de Carrera:** Redacción del presente documento.
  - **Preparación de la presentación:** Realización de la presentación de diapositivas presente para la presentación del Proyecto frente al tribunal.

Para la determinación de las tareas se construye un diagrama *WBS* que representa en una estructura en árbol las tareas y subtareas a realizar de forma ordenada con arreglo a un determinado criterio.

El *WBS* de este Proyecto Fin de Carrera se puede ver en la Figura 8.

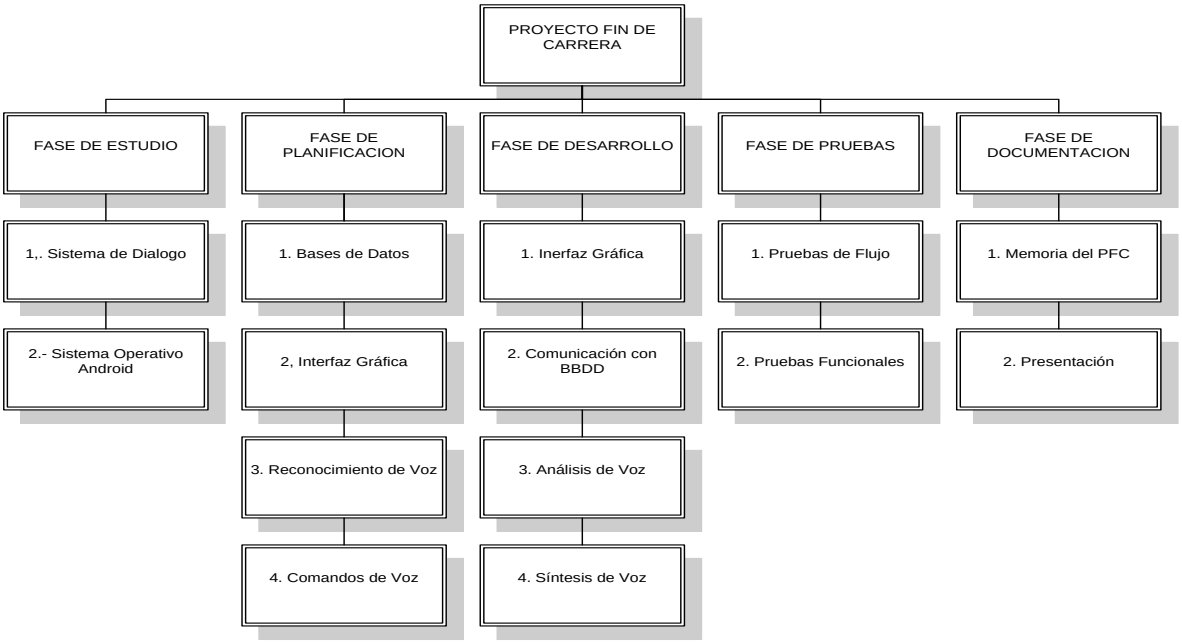
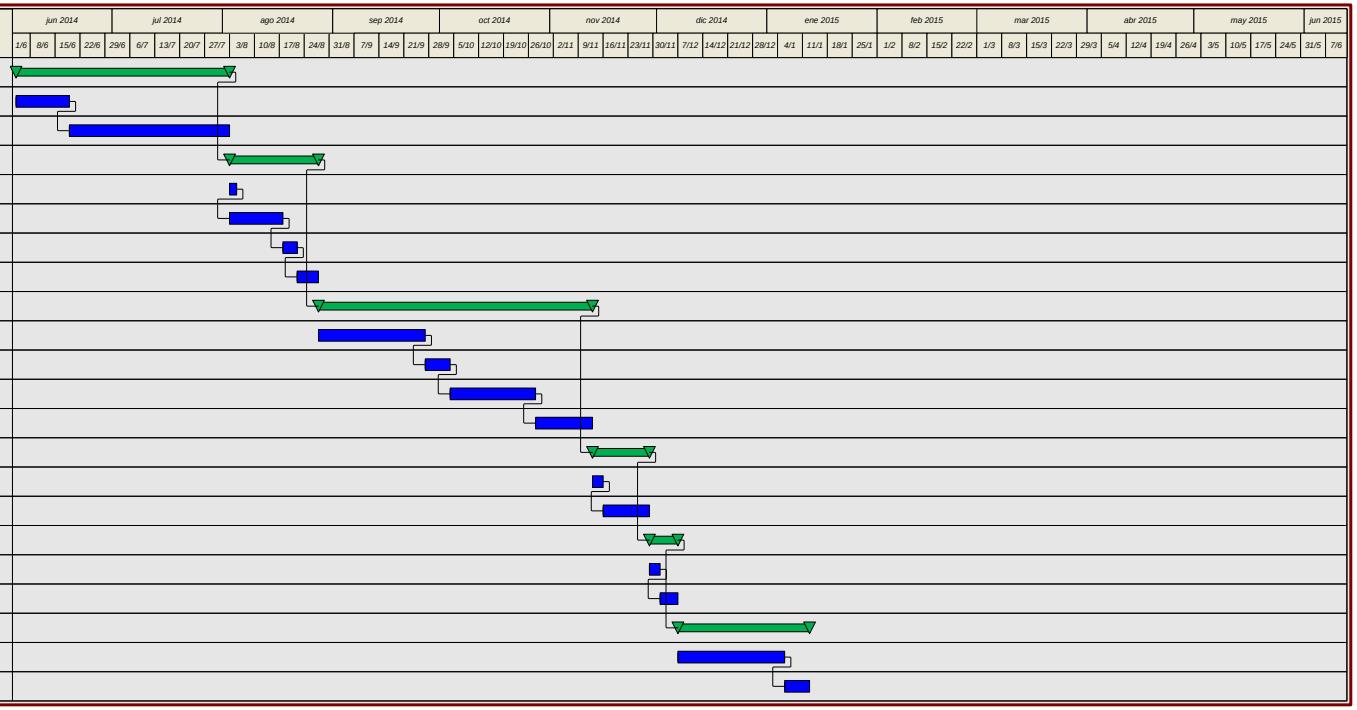



Figura 8. WBS del Proyecto Fin de Carrera

# 1.6 Planificación Temporal

La planificación temporal de acuerdo a la descripción en la sección anterior se resume en el siguiente diagrama:

| ID | Nombre de la Tarea | Comienzo | Final | Duracion |
|---|---|---|---|---|
| 1 | **Fase de Estudio** | **04/06/2014** | **02/08/2014** | **60d** |
| 2 | Sistemas de diálogo | 04/06/2014 | 18/06/2014 | 15d |
| 3 | Sistema operativo Android | 19/05/2014 | 02/07/2014 | 45d |
| 4 | **Fase de Planificación** | **03/08/2014** | **27/08/2014** | **25d** |
| 5 | Bases de Datos | 03/08/2014 | 04/08/2014 | 2d |
| 6 | Interfaz Gráfica | 03/08/2014 | 17/08/2014 | 15d |
| 7 | Reconocimiento de Voz | 18/08/2014 | 21/08/2014 | 4d |
| 8 | Comandos de Voz | 22/08/2014 | 27/08/2014 | 6d |
| 9 | **Fase de Desarrollo** | **28/08/2014** | **12/11/2014** | **77d** |
| 10 | Interfaz Gráfica | 28/08/2014 | 26/09/2014 | 30d |
| 11 | Comunicación con BBDD | 27/09/2014 | 03/10/2014 | 7d |
| 12 | Análisis de Voz | 04/10/2014 | 27/10/2014 | 24d |
| 13 | Síntesis de Voz | 28/10/2014 | 12/11/2014 | 16d |
| 14 | **Fase de Pruebas** | **13/11/2014** | **28/11/2014** | **16d** |
| 15 | Pruebas de flujo | 13/11/2014 | 15/11/2014 | 3d |
| 16 | Pruebas funcionales | 16/11/2014 | 28/11/2014 | 13d |
| 17 | **Fase de Evaluación** | **29/11/2014** | **06/12/2014** | **8d** |
| 18 | Evaluación por parte de usuarios finales | 29/11/2014 | 01/12/2014 | 3d |
| 19 | Evaluación estadística de la aplicación | 02/12/2014 | 06/12/2014 | 5d |
| 20 | **Fase de Documentación** | **07/12/2014** | **12/01/2015** | **37d** |
| 21 | Memoria del Proyecto Final de Carrera | 07/12/2014 | 05/01/2015 | 30d |
| 22 | Preparación de la presentación | 06/01/2015 | 12/01/2015 | 7d |



# Capítulo 2. Estado del arte

Este capítulo servirá para enmarcar el Proyecto Final de Carrera en la situación actual de las nuevas tecnologías, el mundo de los dispositivos móviles y el alcance que tienen estas nuevas tecnologías en el ámbito del usuario.

Se mostrará un estudio de los sistemas de diálogo oral que será el método a utilizar en la comunicación usuario-aplicación y que permitirá el desarrollo de las interfaces de voz del presente proyecto.

Así mismo se introducirá al S.O. Android para dispositivos móviles para conocer el enorme impacto que tiene actualmente en la sociedad mundial y para ello se dará a conocer su desarrollo histórico y la situación actual frente a sus más directos competidores.

## 2.1 Sistemas diálogo oral

### 2.1.1 Definición

La mejor forma de conocer un elemento que se va a estudiar o del que se va a hablar es, entre otras cosas, a partir de su definición. Por ello, comenzaré dando algunas pequeñas pinceladas de lo que se conocen como sistemas de diálogo oral:

- Los sistemas de diálogo basados en procesamiento del habla son sistemas informáticos que reciben como entrada frases del lenguaje natural expresadas de forma oral y generan como salida frases del lenguaje natural expresadas asimismo de forma oral [LOP05]

- Software que acepta lenguaje natural como entrada y produce una salida también oral en lenguaje natural, estableciendo de esta manera una conversación con el usuario. Esta interacción se realiza gracias a todos los procesos y módulos que componen la arquitectura [LOP]

- Los sistemas de diálogo o sistemas conversacionales (SLS, Spoken Language Systems) son una tecnología concebida para facilitar la interacción natural mediante el habla entre una persona y un ordenador. Constituyen como una interfaz o un intermediario entre un usuario y un sistema informático, que tiene la ventaja de no requerir el uso de una

pantalla, un teclado o un ratón y de recurrir, en cambio, al medio de comunicación propio de los seres humanos [LLI06]

### 2.1.2 Características de los sistemas de diálogo

Un sistema de diálogo ideal tiene que ser capaz de:

- reconocer el habla espontánea del ser humano, teniendo en cuenta las diferencias de pronunciación, frecuencia y tono de voz existentes a la hora de exponer la misma frase dos seres humanos con características diferentes;

- comprender los enunciados sin ningún tipo de restricciones de contenido, así como tras interpretar dicho enunciado ser capaz de proporcionar una respuesta que tenga sentido, esté gramaticalmente bien formada y que sea pragmáticamente adecuada;

- al proporcionar la respuesta deber responder con una voz que sea completamente natural y que sea capaz de simular con las características adecuadas a las de un ser humano;

- debe ser un sistema multimodal (Figura 9). Esto significa que el sistema no está limitado a utilizar el habla como único medio de comunicación, tanto en lo que respecta a canal de entrada como de salida, si no que interactúa con otros elementos como canales de comunicación como pueden ser cámaras, pantallas táctiles, gráficos, etc.

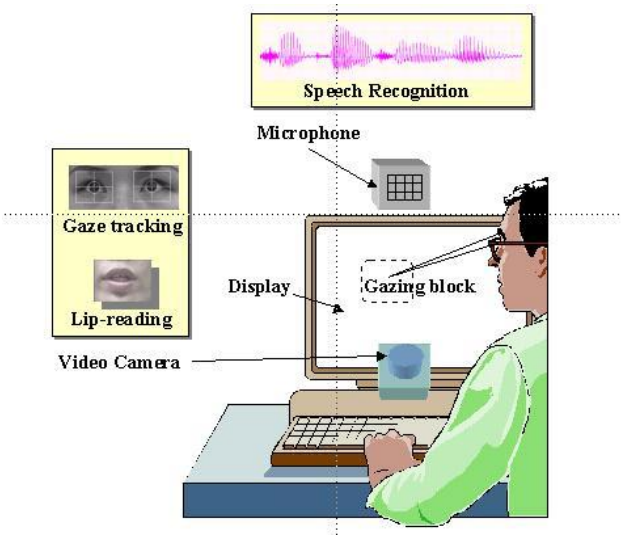



Figura 9: Sistema de diálogo Multimodal

Como ejemplo de sistema de diálogo multimodal podemos ver la película: *"2001, una odisea en el espacio"*, inspirada en la obra de *Arthur C. Clarke* del mismo nombre. En dicha película se puede ver un ordenador de nombre Hal, que responde a todos los patrones que debe tener un sistema de diálogo ideal [PAL].

### 2.1.3 Limitaciones de un sistema de diálogo

La realidad es que los sistemas de diálogo que podemos utilizar hoy en día están sujetos a una serie de limitaciones que, aunque se han ido subsanando y mejorando a lo largo de la historia de estos sistemas, todavía hacen que la distancia hacia el sistema de diálogo ideal con las características nombradas en el apartado anterior de esta memoria sean grandes. De hecho, actualmente estos sistemas:

- están sujetos a las limitaciones del reconocimiento automático del habla;
- poseen una comprensión y respuesta restringidas a dominios específicos;
- están condicionados por la naturalidad del habla sintetizada;
- requieren de estrategias de verificación y corrección de errores;

Pese a este número de limitaciones sí que es cierto que los sistemas de diálogo cada vez se emplean más en numerosas aplicaciones reales, tanto en administración pública como en empresas privadas. Principalmente, y aquí radica su mayor ventaja, es que pueden prestar servicio las 24 horas del día durante 365 días al año, además de mejorar las prestaciones proporcionadas al cliente puesto que se limita el trabajo de los operadores humanos a solucionar problemas más complejos de los que puede resolver con eficacia un sistema automático.

**Limitaciones en el reconocimiento automático del habla.** La principal limitación de los sistemas de diálogo actuales es debida al reconocimiento del habla. Hoy en día no se disponen de reconocedores que puedan procesar de forma adecuada y con el cien por cien de fiabilidad la variación propia del habla completamente espontánea. Ello es debido a:

- **entorno de interacción**. Ruido de fondo, interferencia de dispositivos electrónicos, etc;
- **producción del habla**. Multiplicidad de realizaciones fonéticas de locutores de diversa procedencia geográfica y social: acento, sexo, edad…;

- **familiaridad del usuario con la tecnología**. Usuarios menos expertos con este tipo de sistemas suelen utilizar palabras fuera de vocabulario o fuera del dominio.

**Comprensión y respuesta restringidas a dominios específicos.** Aun suponiendo que pudiéramos contar con los mejores reconocedores del habla, el sistema de diálogo debería ser capaz de comprender cualquier enunciado independientemente del tema o de su complejidad sintáctica, semántica y pragmática. La realidad actual es que los sistemas de diálogo están limitados por lo que se denominan dominios restringidos, es decir, ámbitos concretos a los que se aplica el determinado sistema de diálogo. De este modo, fuera del dominio para el que se utiliza el sistema de diálogo, seguramente deje de ser capaz de interpretar adecuadamente las frases que no están relacionadas con él.

**Naturalidad del habla sintetizada.** Las respuestas del sistema de diálogo para ser ideal no deberían ser muy diferentes con las de un humano, no únicamente siendo gramaticalmente adecuadas, sino que además abarcarán cualquier tema y presentarán las mismas características, en cuanto a riqueza y flexibilidad, que las del lenguaje natural. Para ello, sería necesario un potente y sofisticado sistema de generación del lenguaje que fuera capaz de emular los mecanismos mediante los que las personas conseguimos producir enunciados que respondan acertadamente a cualquier situación, con la tecnología actual, esto todavía no es posible.

Los métodos de conversión de texto en habla, pese a haber experimentado mejoras importantísimas en los últimos años, aún no permiten crear una voz sintetizada que pueda confundirse con la de una persona. Éste es uno de los principales motivos por los que un usuario se percata inmediatamente de que, al llamar a un determinado servicio, quien le responde no es un interlocutor humano, sino un sistema de diálogo.

**Necesidad de estrategias de verificación.** En los sistemas actuales debido a que la interfaz de entrada de voz no siempre es correcta, el propio sistema de diálogo tiene que proveerse de técnicas para verificar que los datos proporcionados se han reconocido y comprendido correctamente. Estas técnicas de verificación se hacen necesarias puesto que un error en el reconocimiento puede hacer que una palabra introducida por el interlocutor sea sustituida por otra diferente o incluso eliminada, de tal forma que el sentido de la frase cambie de forma considerable.

En esta línea existen dos tipos de estrategias de confirmación, explícitas e implícitas [LLI6]:

- **Explicitas**: En los sistemas explícitos el sistema de diálogo genera, lo que podemos denominar, como turno de verificación, en el cual espera a que

el usuario confirme el dato o los datos que acaba de obtener durante el reconocimiento. Este sistema tiene la desventaja, de que, frente a reconocimientos correctos, puede generar turnos de verificación no necesarios que requieren que el usuario interactúe, haciendo el sistema menos eficiente, ya que la confirmación de todos los datos resulta demasiado lenta y repetitiva. Podemos ver un ejemplo de interacción Sistema (S) y Usuario (U), con un sistema de verificación explícita en la Figura 10.

S1: *¿A qué ciudad desea viajar?*
U1: *a Madrid*
S2: *¿Ha dicho que desea viajar a Madrid?*
U2: *sí*
S3: *De acuerdo, destino Madrid. ¿Desde qué ciudad desea salir?*
U3: *desde Granada*
S4: *¿Ha dicho que desea salir desde Ganada?*
U4: *sí*
S5: *De acuerdo, salida desde Granada. ¿Qué día de la semana desea viajar?*
U5: *el lunes*

Figura 10: Ejemplo de confirmación explícita [LLI6]

- **Implícitas**: En este sistema, el sistema de diálogo repite el reconocimiento de los datos, en la siguiente interactuación con el usuario, de forma que los datos únicamente se corrigen en el caso que sea necesario, sin la necesidad de estar confirmándolos antes de la siguiente inserción. La Figura 11 muestra un ejemplo de interacción con confirmación implícita.

S1: *¿A qué ciudad desea viajar?*
U1: *a Madrid*
S2: *De acuerdo, destino Madrid. ¿Desde qué ciudad desea salir?*
U2: *desde Granada*
S3: *De acuerdo, salida desde Granada. ¿Qué día de la semana desea viajar?*
U3: *el lunes*

Figura 11: Ejemplo de confirmación implícita [LLI6]

### 2.1.4 Arquitectura de los sistemas de diálogo

Los sistemas de diálogo se diseñan con una estructura modular. Esto es, están separados en bloques independientes, de forma que cada cual realiza su tarea de forma individual, y presenta su salida como entrada al modulo siguiente. El resultado de la interactuación de todos los módulos da lugar al funcionamiento del sistema completo, en este caso, un sistema de diálogo. En la Figura 12 tenemos el esquema de funcionamiento de un sistema de diálogo en el que se puede ver claramente la estructura modular descrita.

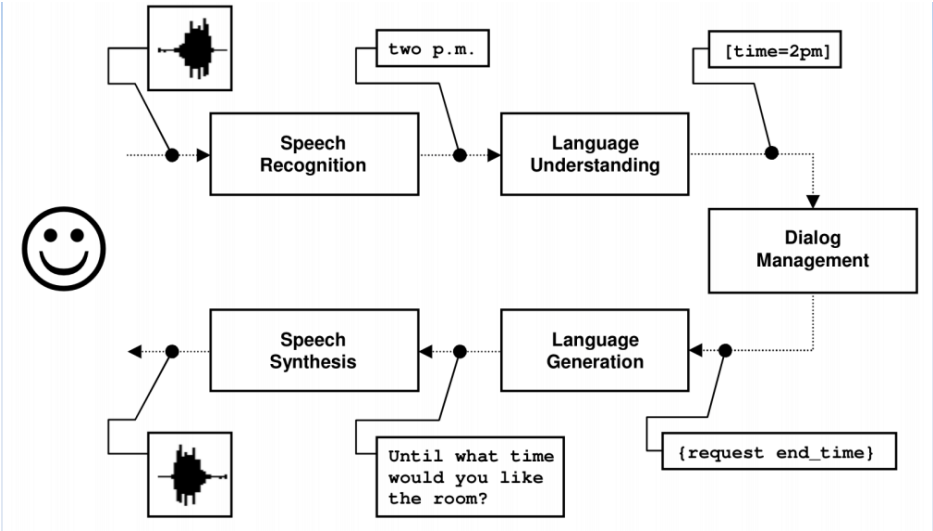



Figura 12: Estructura modular de los sistemas de diálogo

## 2.1.5 Reconocimiento del habla

El proceso de Reconocimiento Automático del Habla (RAH) es el encargado de, partiendo de una entrada oral pronunciada por el usuario, transfórmala en una secuencia de palabras.

Por tanto, el objetivo principal de este módulo es transformar una onda sonora a texto (Figura 13). La entrada del módulo no es más que una onda que tiene sus características frecuenciales, así como una amplitud. Esta forma de onda, depende (entre otras cosas), de los fonemas que ha pronunciado el usuario y que forman la frase que debe ser reconocida. El módulo a través del procesamiento de esta señal, y teniendo en cuenta que en el lenguaje natural, cada fonema, letra, palabra tiene sus componentes frecuenciales, debe ser capaz de obtener el texto que representa la onda sonora.


Figura 13: Transformación de las señales acústicas a texto

Es un proceso complejo que se basa en obtener un conjunto de patrones que pueden ser comparados con la señal de entrada, que es la que pretende ser reconocida, y obteniendo a la salida del módulo aquellos patrones que **con mayor probabilidad** representan dicha señal de entrada (Figura 14).


Figura 14: Comparación de patrones en el módulo RAH

Los problemas a los que se enfrenta este módulo para ser completamente eficaz son:

- **Variaciones de pronunciación de los hablantes**. No suenan igual los sonidos generados por diferentes personas.
- **Ambigüedades acústicas**. En ocasiones la misma señal acústica se puede ajustar a patrones diferentes. Son palabras que se pronuncian igual pero de escriben de forma distinta. Ej: vaca y baca.
- **Variaciones de producción**. Variaciones que producimos de forma involuntaria los seres humanos y que hacen que el sonido se aleje de su forma ideal:
- **Falta de cuidado en la pronunciación de palabras:** En ocasiones pronunciamos con demasiada velocidad u omitimos palabras de breve duración incluso combinándolas con las siguientes.
- **Variación en la pronunciación:** Los estados de ánimo son causa en la variación de la pronunciación y entonación de sonidos.
- El **contexto** en el que se produce el sonido puede llevar a la necesidad de diferentes patrones de acuerdo a la situación o el entorno en el que se realiza el reconocimiento para un mismo sonido.

Debido a estos problemas el modulo RAH está basado decisiones, incluso muchas veces a pesar de falta de información y con la seguridad de que no todas las decisiones resultarán correctas. Esto supone la necesidad de utilizar alguna técnica para tomar dichas decisiones, incluso si fuera necesario, alguna técnica de corrección de errores.

Una de las técnicas más utilizadas a tal fin está basada en la "*La teoría de la Decisión Estadística*", la cual consiste en valorar distintas secuencias de

palabras y después tomar como válida la que tenga mayor probabilidad de estar asociada a la secuencia acústica de entrada.

De esta forma podemos aplicar algunos criterios de clasificación para los módulos RAH atendiendo a sus características y funcionalidades [GRI07]:

- **Iniciativa del diálogo**
  - **Sistemas de diálogo guiados por el sistema.** En estos sistemas existe una interacción entre el sistema y el usuario que destaca por ser un sistema de turnos fijados entre preguntas y respuestas.
  - **Sistemas de iniciativa mixta**. Aceptan la existencia de interrupciones por parte del usuario, repartiendo de forma equilibra el turno de palabra. Además incorporan mecanismos de detección de incoherencias gramaticales.
  - **Sistemas de diálogo guiados por el usuario**. El usuario tiene la iniciativa del diálogo. De forma que al usuario se le permite el uso del lenguaje natural y está delimitado por los mensajes que ofrece el sistema.
- **Dependencia del hablante**
  - **Dependiente del hablante:** Este sistema está preparada para interactuar únicamente con un hablante. Este tipo de sistemas son más fáciles de implementar, baratos y precisos, pero tienen poca flexibilidad.
    - **Independientes del hablante**: Esta desarrollado para cualquier hablante de un determinado tipo (por ejemplo, todas aquellas personas que tienen el mismo idioma). Son complicados de desarrollar, más caros y menos precisos.
  - **Tipo de comunicación**
    - **Unimodal:** Únicamente se utiliza el habla para comunicarse con el sistema.
    - **Multimodal:** Utilizan diferentes canales de comunicación para la entrada de datos. Como por ejemplo: el habla, teclado, ratón, cámara….
  - **Tipo de discurso**
    - **Reconocimiento de palabras aisladas.** En este tipo de reconocimiento el patrón acústico de las palabras del vocabulario se almacena como una secuencia temporal de características derivadas, usando LPC, análisis de banco de filtros o alguna otra

técnica de análisis del lenguaje. El reconocimiento se realiza comparando la diferencia entre el patrón acústico analizado y los patrones almacenados, obteniendo como resultado del reconocimiento la palabra que sea menos diferente.

- **Reconocimiento de palabras conectadas.** En este caso la entrada hablada es una secuencia de palabras de un vocabulario específico y el reconocimiento se lleva a cabo basándose en la coincidencia de palabras de referencia aisladas.

- **Reconocimiento continuo.** Un sistema de este tipo funciona sobre un lenguaje en el que las palabras están conectadas (no están separadas por pausas). Este tipo de reconocimiento tiene complicaciones del tipo:

    - Dificultad para reconocer el comienzo y el final de las palabras.

    - La generación de cada fonema se ve afectada por la generación de los fonemas adyacentes, y de modo parecido el comienzo y el final de las palabras se ven afectados por las palabras que preceden y suceden.

    - Frecuencia del habla. En el caso de un discurso más rápido se ve dificultado el reconocimiento.

- **Reconocimiento discreto.** Funciona con palabras simples, con la necesidad de que exista una pausa entre la dicción de cada palabra. Este es el tipo de reconocimiento más sencillo, puesto que los puntos de finalización son más fáciles de encontrar y la pronunciación de una palabra no afecta.

- **Adaptación.** En un sistema adaptativo, el propio sistema es capaz de aprender nuevas estrategias comunicativas en función del comportamiento del usuario.

### 2.1.6 Procesamiento del Lenguaje Natural

Utilizando como entrada el resultado del módulo de reconocimiento del habla, esta etapa del sistema de diálogo extrae el significado de las palabras reconocidas en la frase expresada por el interlocutor humano, generando la representación semántica de la frase expresada en términos de un lenguaje especificado para la tarea. Este módulo sigue la siguiente estructura:

- **Análisis morfológico-léxico.** Permite dividir las palabras en lexemas y morfemas, de forma que se obtienen las distintas clases de palabras.

- **Análisis sintáctico.** Obtiene la estructura jerárquica de las frases y extrae el significado de la estructura sintáctica compleja obtenida a partir del significado de sus constituyentes.
- **Análisis semántico.** Obtiene un pre-procesamiento del significado.
- **Análisis pragmático.** Añade información al análisis semántico de la frase en función del contexto de la interacción en el que aparece.

En la Figura 15 se muestra un ejemplo de comunicación entre un usuario y un sistema de diálogo que gestiona la información de horarios de trenes. El sistema tiene que ser capaz a partir de una frase introducida por el usuario de realizar un análisis de la misma, buscando palabras clave de forma que posteriormente esto le sirve para interpretar y comprender la información que se le está solicitando.

Eh, pues mire, quería saber a qué hora sale la… el último tren, eh… hacia Barcelona, desde Madrid, el sábado

PETICIÓN
DESTINO="Barcelona"
ORIGEN="Madrid"
FECHA:
DÍA="28"
HORA="Último"

Figura 15: Reconocimiento de Voz. Adquisición y entrenamiento

Si tenemos en cuenta que una persona puede leer fácil y rápidamente un texto con un asombroso porcentaje de letras mal escritas, permutadas u omitidas. Esto nos hace darnos cuenta que en el lenguaje humano debe existir uno o varios patrones subyacentes que se repiten de forma frecuente y que permiten crear mecanismos para la reconstrucción de datos escritos que permiten eliminar y acotar estos errores. Es en esta idea en la que se basa el módulo del procesamiento del lenguaje natural.

### 2.1.7 Gestión del diálogo

El gestor del diálogo gestiona la interacción entre la persona y el sistema de diálogo para la realización de la tarea deseada mediante la coordinación de las funciones desarrolladas por el resto de módulos. Es, de este modo, el módulo central en un sistema de diálogo que se encarga de permitir la coherencia entre las preguntas que realiza el usuario y las respuestas que aporta el sistema a dichas preguntas. Incluso en base a la secuencia que sigan dichas preguntas y respuestas puede ser capaz de adelantarse a las reacciones del usuario.

Es el módulo que decide qué paso debe dar el sistema tras cada intervención del usuario, posibilitando que la interacción con el usuario sea lo más cómoda e "inteligente" posible. Para ello, se basa en la interpretación semántica de la

petición del usuario, la historia previa del diálogo, la información disponible en ese punto, el estado actual del sistema, la información obtenida de la base de datos, la estrategia definida, etc.

Un gestor del diálogo debe cumplir las funciones siguientes [LAR03]:

- actualizar el contexto del diálogo;
- proveer de contexto en el que basar las interpretaciones;
- coordinar el resto de módulos del sistema;
- decidir qué información dar al usuario, y cuándo hacerlo.

De esta forma, este módulo coordina además todas las fuentes de información del sistema: los resultados de procesar el lenguaje natural, las consultas de bases de datos, la obtención y consulta de conocimientos acerca del dominio de la aplicación, las intervenciones de los usuarios, así cómo archivar y ser capaz de “aprender” de la historia previa del diálogo.

### 2.1.8 Generación del Lenguaje Natural

A través del proceso de Generación del Lenguaje Natural (GLN) se obtiene la construcción de expresiones o partes de ellas de cualquier tamaño tanto habladas como escritas en lenguaje natural a partir de una representación no lingüística. Así, podemos decir que este modulo traduce la acción seleccionada por el gestor de diálogo en una frase en lenguaje natural que será comunicada al usuario.

El sistema debe escoger la información de las bases del conocimiento, decidir cómo organizarla, y determinar cómo producir el texto en lenguaje natural, incluyendo la decisión de la entrada léxica y la estructura sintáctica. Este proceso se lleva a cabo mediante la organización del contenido, distribución del contenido en fases, la lexicalización del contenido, generación de expresiones referenciales y realización lingüística. Es importante optimizar el texto para obtener mensajes legibles, adaptando el vocabulario y la complejidad de las estructuras sintácticas a la destreza lingüística del usuario [LLI06].

En la figura 16, podemos ver que una vez, se ha realizado la compresión de la petición del usuario, el siguiente paso, es consultar probablemente una base de datos para extraer la información que ha sido solicitada y generar luego el texto que corresponderá con la respuesta a la información solicitada.

Petición
DESTINO="Barcelona"
ORIGEN="Madrid"
Fecha:
DÍA="28"
HORA="Último"

El último tren sale a las 11 y 53 minutos de la noche

Figura 16: Generación del texto de respuesta

## 2.1.9 Síntesis de Texto a Voz

Un sistema de síntesis oral se encarga de transformar un texto escrito en una señal acústica. Los sintetizadores de texto a voz se componen de dos fases fundamentales, una encargada de convertir el texto plano, compuesto por símbolos y abreviaturas, en sus palabras asociadas, o lo que es lo mismo, se encarga de normalizar el texto y de asignar transcripciones fonéticas a las palabras dividiendo y marcando el texto en frases, cláusulas y oraciones; y la segunda encargada de convertir la representación lingüística simbólica en sonidos, utilizando diversas técnicas para simular los sonidos producidos por las cuerdas vocales humanas. El objetivo principal de este modulo queda descrito visualmente en la Figura 17 [DUT97].


Figura 17: Objetivo del módulo TTS

En la figura 18, se pueden apreciar los diferentes módulos englobados en cada una de las fases del sistema de conversión de texto en habla descritos anteriormente.

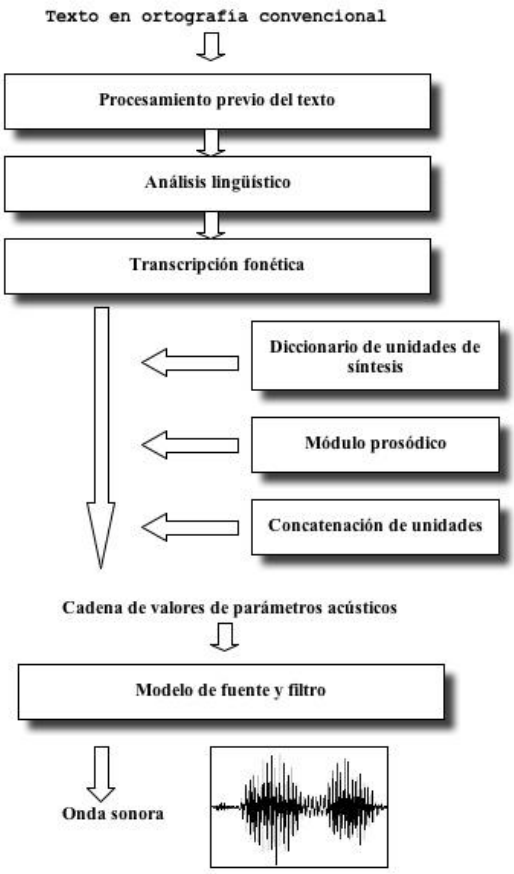



Figura 18: Módulos de un sistema de conversión de texto en habla

# 2.2 Sistemas móviles

En la actualidad nuestra vida cotidiana está marcada por la utilización de un sinfín de dispositivos móviles. Este tipo de dispositivos abarcan un amplio mercado que va desde dispositivos reproductores multimedia a teléfonos móviles o tablets de última generación.

El boom de los dispositivos móviles tiene lugar principalmente en el mundo de los videojuegos cuando empresas como Nintendo o Sony empezaron a comercializar pequeños dispositivos que podían trasladarse a cualquier lugar para ser utilizados.

## 2.2.1 Características de los sistemas móviles

Lo que entendemos por dispositivo móvil principalmente es aquel que cumple con las siguientes características:

- Son aparatos pequeños;
- con capacidad de procesamiento;

- con conexión permanente o intermitente a una red;
- con memoria limitada;
- diseñados específicamente para una función concreta, pero que pueden llevar a cabo otras más generales;
- normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, la cual puede adaptarlo a su gusto.

Dadas este conjunto de características se puede pensar que estos dispositivos se pueden entender hasta cierto punto como una computadora tradicional. Sin embargo, nada más lejos de la realidad, aunque estos dispositivos cada vez tienden más a acercarse a ser pequeñas computadoras, aun tienen ciertas diferencias que los hacen alejarse de ellas. Estas diferencias son [TIP14]:

- Tienen funcionalidad limitada;
- no son necesariamente extensibles y actualizables;
- en pocos años el usuario debería cambiarlo;
- menos complicados en su manejo;
- fácil aprendizaje en su operación;
- no se requieren usuarios especializados.

### 2.2.2 Aplicaciones móviles

Otra de las revoluciones que trajeron los sistemas móviles, es que en la mayoría de los casos su utilización es a través de lo que denominamos aplicaciones.

Inicialmente estas aplicaciones estaban principalmente destinadas a una función puramente recreativa. Sin embargo con el tiempo el mercado de las aplicaciones móviles despertó un gran interés entre los usuarios permitiendo que se ampliar en gran medida en sus diferentes funcionalidades tales como: registro de datos, manuales de diferentes operaciones, herramientas de trabajo, etc. Esto ha desembocado en un sinfín de tipos de aplicaciones que demuestran el potencial de los dispositivos móviles en que puedan ir instaladas dichas aplicaciones.

### 2.2.3 Principales Sistemas Operativos Móviles

Los sistemas móviles por sus características especiales que ya hemos descrito requieren de sistemas operativos propios y adaptados a sus capacidades. No obstante dentro de los sistemas operativos móviles existe gran diversidad de opciones para cada dispositivo, lo que da lugar a una alta

competencia y a la existencia de un gran número de sistemas operativos móviles.

Aquí describiré los más importantes del mercado y daré una breve pincelada sobre cada uno de ellos:

- **iPhone OS/iOS (Apple):** El sistema operativo creado por la marca de Cupernico y que en primer lugar fue desarrollada únicamente para su dispositivo iPhone y denominándose iPhone OS. En la actualidad el principal sistema operativo de la compañía se denomina iOS y es el compartido por la gran mayoría de sus dispositivos móviles. Se trata de un sistema operativo desarrollado íntegramente por el equipo de Apple y no está permitida su instalación en dispositivos de terceros.

- **Android OS (Google Inc.**): Este sistema operativo para móviles es, en la actualidad, el preferido por la gran mayoría de usuarios. Se trata de un sistema operativo de código libre y abierto lo que facilita que existan en el mercado una gran cantidad de apps gratuitas y de bajo coste. Además de que las actualizaciones de dicho sistema operativo sean muy frecuentes. Implementa una máquina virtual de Java especial llamada 'Dalvik' la cual es la encargad de ejecutar todas las aplicaciones en el dispositivo.

- **Blackberry OS:** Es un sistema operativo móvil cerrado y desarrollado por la empresa "Research in Motin". Fue desarrollado para el uso exclusivo del fabricante de terminales 'Blackberry'. Dicho fabricante está especializado en dar servicio a grandes corporaciones y ofrece a los usuarios una gran capacidad de sincronización con conocidas herramientas de trabajo tales como: Microsoft Exchange, Lotus Domino, Novell GrupWise, etc.

- **MeeGo Os (Nokia e Intel):** Ambos fabricantes se unieron para obtener este sistema operativo de código libre y abierto. Está basado en dos tecnologías, la primera pertenece a Nokia de nombre Maemo y la segunda perteneciente a Intel y que recibe el nombre de Moblin. Éste sistema operativo está diseñado para una cierta arquitectura de dispositivos, la ARMv7, utilizando microprocesadores Intel Atom.

- **Bada (Samsung Electronics):** Bada es un sistema operativo del cual es propietario Samsung que fue lanzado al mercado por primera vez en 2010 y que integra actualmente en algunos de sus dispositivos. Es un sistema operativo que incluye novedades como gráficos 3D y gestos 'multi-touch'.

- **Symbian OS (Nokia):** Symbian es un sistema operativo móvil muy ligado a los teléfonos que ofrecen una gran integración con comunicaciones y la

administración de información personal. Combina el ‘middleware’ con las comunicaciones inalámbricas con integración al buzón de entrada del correo electrónico. Nokia no mantiene Symbian como un proyecto libre de código abierto.

- **Web OS (Palm/HP):** Es un sistema operativo para móviles que corre una versión del kernel Linux. WebOS fue inicialmente desarrollado por Palm como sucesor de su sistema operativo PalmOS 5. Es un sistema operativo propietario de Mobile OS que fue adquirida por HP. Hoy en día es el sistema operativo más utilizado en la gama de productos HP.

- **Windows Mobile:** Es un sistema operativo móvil cuyo propietario es Microsoft. Está usado en Smartphone y dispositivos móviles con o sin pantalla táctil. Basado en el kernel Windows CE 5.2. En 2010 Microsoft anunció una nueva plataforma para dispositivos a la que llamaron Windows Phone 7.

# 2.3 Android

El marco de este Proyecto Fin de Carrera está enfocado en uno de los sistemas operativos móviles más extendidos en la actualidad. Su nombre es Android. En los próximos apartados de esta memoria me dedicaré a dar a conocer este sistema operativo móvil.

## 2.3.1 Definición

Android es un sistema operativo que inicialmente fue diseñado para utilizarse en dispositivos móviles. Sin embargo en la actualidad este sistema operativo se utiliza en múltiples dispositivos, tales como tablets, GPS, televisores, discos duros multimedia e incluso en electrodomésticos convencionales [PEA].

Android está basado en una versión modificada del núcleo de Linux, que se trata de un sistema operativo libre, gratuito y multiplataforma. De forma que Android se trata de un sistema abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones. Éstas pueden ser reemplazadas libremente, además desarrollarlas por terceros, a través de herramientas proporcionadas por Google, generalmente utilizadas a través de un lenguaje de programación que resulta una variación de Java llamado Dalvik, y que proporciona todas las interfaces necesarias para desarrollar fácilmente aplicaciones que acceden a las funciones del teléfono (GPS, llamadas, agenda, etc.).

El código fuente de Android está disponible bajo diversas licencias de software libre y código abierto. Google liberó la mayoría del código de Android bajo la licencia Apache. Esta licencia permite que un desarrollador no sólo pueda modificar su código, sino también mejorarlo. A través de esas mejoras puede publicar el nuevo código y ayudar a mejorar el sistema operativo en futuras versiones. Esta es una de las causas principales de que existan cientos de miles de aplicaciones disponibles, que extienden la funcionalidad de los dispositivos y mejoran la experiencia del usuario.


Figura 19. Logotipo Android

## 2.3.2 Historia

La historia de Android [GIR] comienza en el año 2003 en la ciudad de Palo Alto en el estado de California de la mano de Andy Rubin, Rich Miner, Nick Sears y Chris White cuando crearon una pequeña empresa de nombre Android Inc. Su objetivo inicial era la de desarrollar un sistema operativo inteligente, que tuviese en cuenta la localización y gustos del usuario y que utilizara esta información para actuar en consecuencia. Dos años más tarde, en agosto de 2005 Google se interesa por el trabajo de la empresa y adquiere la compañía para tomar partido en el desarrollo.

En el año 2007 la salida al mercado del iPhone fue un duro golpe para el proyecto que ya entonces desarrollaba Google. Puesto que en ese momento contaban con un prototipo de sistema operativo que funcionaba en un teclado de tipo QWERTY, muy similar al que utilizaban las BlackBerry. De forma que tuvieron que cambiar el rumbo para adaptarlo a un dispositivo móvil con pantalla táctil para poder competir con su competidor.

Ese mismo año, en concreto el 5 de Noviembre de 2007 se da a conocer la fundación Open Handset Alliances que se trataba de un consorcio de numerosas compañías de hardware, software y telecomunicaciones (Google, Sony, HTC, Samsung, T-Mobile, Sprint Nextel, Texas Instruments, etc.) comprometidas con la promoción de estándares abiertos para dispositivos móviles. Ese mismo día Android presenta su primer producto: Un Sistema

Operativo basado en Linux 2.6 junto con el SDK (Kit de desarrollo de software) que permitiría que los desarrolladores pudieran comenzar a crear sus aplicaciones para el nuevo sistema operativo. Inicialmente tuvo un desarrollo lento puesto que se lanzo antes el sistema operativo que el primer terminal móvil que lo utilizara.

El 22 de Octubre de 2008 sale al mercado el primer Smartphone con el sistema operativo Android. Se trataba del HTC Dream, vendiéndose en apenas 6 meses más de 1 millón de terminales en EEUU. Esta primera versión del sistema operativo recibió el nombre de “Android 1.0 Apple Pie”.

### 2.3.3 Historial de actualizaciones y versiones

Desde sus inicios Android ha sufrido numerosos cambios hasta llegar a la actualidad. Este es un proceso normal dentro de los sistemas operativos, puesto que estas actualizaciones del sistema sirven típicamente para arreglar defectos o añadir nuevas funciones

A partir de sus inicios en que su primera versión se denomino “Android 1.0 Apple Pie”, las nuevas actualizaciones del sistema operativo han sido nombradas siempre en orden alfabético y recibiendo nombres de postres escritos en ingles.

De esta forma, hasta llegar a la actualidad han correspondido las siguientes versiones del Sistema Operativo:

- **Apple Pie (Versión 1.0).** Primera versión del Sistema Operativo móvil totalmente gratuito y Open Source frente a su directo competidor iOS. Presentado oficialmente el 23 de Septiembre de 2008. Sus principales características eran:
    - menú desplegable de notificaciones;
    - widgets de escritorio;
    - android Market. La tienda para conseguir apps que inicialmente no tenía ningún soporte para pago de aplicaciones, inicialmente todas eran gratuitas;
    - integración con Google Mail, Contacts y Calendar;
    - navegador, Maps, Google Talk, reproductor de Youtube y soporta para cámaras.


Figura 20. Versión 1.0 Android. *Apple Pie*

- **Banana Bread (Versión 1.1).** Aparece en febrero de 2009 y, aunque no añade grandes novedades, sí que corrige numerosos fallos y bugs. Pero sobre todo incluye un concepto innovador que hasta entonces era poco usado: las actualizaciones automáticas, de forma que resultaba muy simple mantener todo el software del terminal al día.

- **Cupcake (Versión 1.5).** Lanzada al mercado el 30 de abril de 2009, esta nueva versión del sistema operativo tenía las siguientes características:
  - nuevo teclado táctil QWERTY en pantalla con predicción de texto;
  - capacidad de subir videos a Youtube e imágenes a Picasa directamente desde el teléfono;
  - cam Recorder para grabación y reproducción de videos;
  - widget de escritorio de google para búsquedas;
  - SDK para desarrollo de widgets por parte de terceros;
  - soporte para Bluetooth A2DP y AVRCP;
  - transiciones de pantalla animadas.


Figura 21. Versión 1.5 Android. *Cupcake*

- **Donut (Versión 1.6).** Aparece en Septiembre de 2009. En esta ocasión la interfaz sufrió pequeños retoques visuales, pero lo realmente nuevo estaba incluido en el núcleo del S.O.:
  - soporte para CDMA/EVDO, 802.1x, VPN;
  - búsqueda por voz actualizada, con respuesta más rápida y mayor integración con aplicaciones nativas, incluyendo la posibilidad de marcar a contactos.
  - compatible para distintas resoluciones de pantalla. Soporte WVGA;

- nuevo diseño del Android Market;
- utilidad de búsqueda universal en Internet y en el mismo dispositivo;
- rediseño de la interfaz de la aplicación cámara de fotos.


Figura 22. Versión 1.6 Android. *Donut*

- **Eclair (Versión 2.0 y 2.1):** Liberada en Octubre del 2009. Fue una de las actualizaciones más sustanciales sufridas por Android tanto a nivel de diseño como de arquitectura interna. Era una versión dirigida a dispositivos de mayor tamaño en un tiempo en el que los fabricantes empezaban a diversificar su oferta. Sus mayores novedades fueron:
  - soporte para múltiples cuentas de usuario;
  - Google Maps Navigation, es decir, sistema de navegación GPS gratuito;
  - compatible con Microsoft Exchange;
  - soporte para múltiples pantallas;
  - navegador de internet actualizado: soporte para HTML5 y barra de dirección y búsqueda unificada;
  - función Text to Speech para escribir textos mediante el uso de voz;
  - nueva pantalla de desbloqueo;
  - zoom digital para la cámara.


Figura 23. Versión 2.0 Android. *Eclair*

- **Froyo (Versión 2.2):** El de 20 mayo de 2010 aparece esta nueva actualización del S.O., muchos de ellos orientados al uso empresarial. Cabe destacar que el primer dispositivo móvil que fue actualizado fue

el Nexus One precisamente lanzado por Google, lo que dejaba entrever cuál sería la política de la empresa desde entonces:

- pantalla de inicio totalmente rediseñada con 5 paneles en lugar de 3;
- nueva galería de imágenes;
- soporte para actuar como *hotspot* para otros dispositivos, *thetering* de datos;
- soporte para Flash 10.1;
- función copiar y pegar mejorada en Google Mail;
- nueva pantalla alternativa de desbloque mediante código PIN;
- nuevo compilador que mejora la velocidad;
- grabación de video en alta calidad a 720p;
- compilador JIT;
- motor JavaScript V8 de Chrome usado en el Browser.


Figura 24. Versión 2.2 Android. *Froyo*

- **GingerBread (Versión 2.3):** A finales del año 2010, concretamente en Diciembre Google lanza esta nueva revisión que se convertiría en la versión de Android más extendida. Sus nuevas características fueron:
    - revisión estética completa: pantalla de inicio, barra de estado, etc;
    - nuevo diseño para el teclado numérico en pantalla;
    - compatible con pantallas y resoluciones mayores;
    - función para copiar y pegar mejorada con soporte para caracteres individuales en lugar de cajas de texto;
    - teclado en pantalla mejorado;
    - soporte para NFC;
    - herramientas para visualización de consumo y uso de batería mejoradas;
    - soporte para cámaras frontales;
    - acceso de bajo nivel para los desarrolladores de juegos;

- sustitución del sistema de archivos YAFFS por ext4.


Figura 25. Versión 2.3 Android. *Gingerbread*

- **Honeycomb (Versiones 3.0/3.1/3.2):** Se trató de una actualización específica para tablets, no compatible con teléfonos. Aparece en febrero de 2011 con las siguientes características principales:
    - pantalla de inicio rediseñada;
    - inclusión de tonos azules en la interfaz en detrimento del verde habitual;
    - nuevas funcionalidades para el emplazamiento y uso de widgets;
    - fin de los botones físicos. Adaptación automática del SO según el dispositivo;
    - multitarea mejorada;
    - aceleración gráfica mediante hardware;
    - optimización del renderizado de gráficos 3D;
    - video chat en Google Talk;
    - soporte para periféricos USB.


Figura 26. Versión 3.0 Android. *Honeycomb*

- **Ice Cream Sandwich (Versión 4.0):** Es una versión basada en HoneyComb pero optimizada para *Smartphone*. Lanzada al mercado el 11 de Octubre de 2011 en poco tiempo se convierte en el sistema operativo móvil más usado del mundo. Sus principales novedades fueron:
    - utiliza una nueva fuente tipográfica de nombre “Roboto”;
    - nueva Interfaz gráfica de nombre “Holo”;
    - sistema de gestión de notificaciones mejorado;

- multitarea mejorado;
- sugerencias y diccionarios para el teclado virtual;
- nuevo diseño y funcionalidades para la pantalla “Home”;
- funcionalidad para transferir datos entre dos dispositivos NFC;
- función de desbloque mediante el rostro;
- nuevas funciones para visualización y gestión del consumo de datos;
- nuevas aplicaciones de correo y calendario;
- soporte MKV;
- soporte lápiz táctil.


Figura 27. Version 4.0 Android. *Ice Cream Sandwich*

- **Jelly Bean (Versión 4.1/4.2/4.3):** En julio de 2012 Google presenta esta nueva actualización con los siguientes cambios:
  - desaparece el soporte para Flash Player;
  - rendimiento del sistema y gráfico mejorado;
  - sistema de detección de entrada de datos táctiles optimizado;
  - estreno de Google Now, el servicio-asistente de voz inteligente de Google;
  - navegador Google Chrome;
  - búsqueda mediante voz mejorada;
  - rediseño de la tipografía “Roboto”;
  - nuevas posibilidades para notificaciones interactivas de escritorio;
  - mejoras en el ajuste de tamaño de los widgets de escritorio;
  - mejoras en la corrección ortográfica y la predicción del teclado;
  - dictado de voz offline.


Figura 28. Versión 4.1 Android. *Jelly Bean*

En Octubre de 2012 llegó Android 4.2 que sin embargo mantuvo el nombre de “Jelly Bean” con las novedades:

- rendimiento mejorado;
- nuevas animaciones;
- nuevo panel de control;
- acceso a widgets y cámara fotográfica desde la pantalla de bloqueo;
- soporte para función de streaming de video y audio desde el terminal;
- rediseño de la aplicación reloj y sus widgets;
- soporte para varios perfiles de usuario;
- captura de fotografías panorámicas de 360º;
- gestual Mode para personas invidentes.

El 24 de Julio de 2013 Google anuncia Android 4.3 que continuaba con el mismo nombre. Esta nueva versión pretendía consolidar Android como un sistema operativo capaz de hacer correr juegos. Sus novedades fueron:

- soporte multiusuario y de perfiles mejorado;
- soporte para gráficos OpenGL ES 3.0;
- compatible con TRIM;
- bluethooth Smart;
- plataforma Google Games;
- servicios de localización Wi-Fi mejorados.

- **KitKat (Versión 4.4):** Es la última versión de Android hasta el momento. Esta nueva versión corrige uno de los principales defectos que tiene el Sistema Operativo: hay muchas versiones y los fabricantes tienen dificultades para adaptar sus productos a los requisitos de las últimas versiones, lo que hace que muchos usuarios no se actualicen. Las novedades que trae este sistema operativo han sido:
    - rebaja de requisitos hardware para corregir la fragmentación de versiones;

- compatible con terminales con 512 Mb de memoria RAM;
- reducción del consumo de batería mediante la optimización de sensores;
- incluye la herramienta de ofimática QuickOffice;
- servicios de almacenamiento online integrados: Google Drive, Box, etc.;
- soporte para infrarrojos;
- aplicaciones a pantalla completa, modo inmersivo;
- soporte Bluetooth HID y Bluetooth Message Access Profile;
- captura de pantalla en video;


Figura 29. Versión 4.4 Android. *Kitkat*

En el momento de escribirse esta memoria las cuotas de mercado de cada una de las versiones de Android eran las mostradas en la Figura 30 [XAT14].

| Version | Codename | API | Distribution |
|---|---|---|---|
| 2.2 | Froyo | 8 | 0.8% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 14.9% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 12.3% |
| 4.1.x | Jelly Bean | 16 | 29.0% |
| 4.2.x | | 17 | 19.1% |
| 4.3 | | 18 | 10.3% |
| 4.4 | KitKat | 19 | 13.6% |

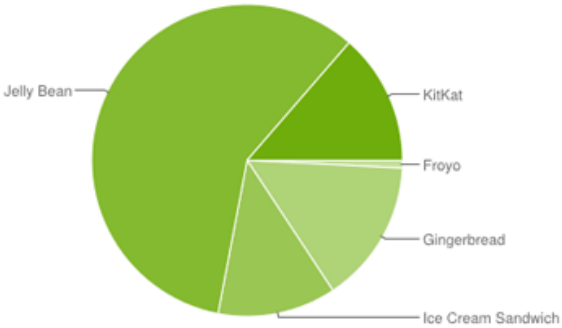



Figura 30. Cuota de mercado de cada una de las versiones de Android

## 2.3.4 Principales características [JES]

En la actualidad encontramos un gran número de plataformas para dispositivos móviles (iOS, Symbian, Windows Phone…). Sin embargo, ¿Qué es lo que hace a Android especial respecto a estos otros sistemas? La respuesta es que es el primero capaz de combinar en una misma solución las siguientes características:

- **Plataforma abierta.** Probablemente la característica más destacada de Android es que es una plataforma de desarrollo libre basada en Linux y de código abierto.

- **Adaptable a cualquier tipo de hardware.** Ya que está diseñado para ser utilizado en un amplio tipo de dispositivos, no únicamente en móviles o tabletas. Sin embargo esto también resulta un hándicap para los desarrolladores puesto que una aplicación correcta debería funcionar en dispositivos con gran variedad de tipos de entrada, pantalla, memoria, etc.

- **Portabilidad.** Para el desarrollo de sus aplicaciones se utiliza el lenguaje de programación Java lo que nos asegura poder ejecutar las aplicaciones finales en cualquier tipo de CPU gracias al concepto de máquina virtual.

- **Gran cantidad de servicios incorporados.** Por citar alguno de los muchos ejemplos podemos nombrar el reconocimiento y síntesis de voz, navegador, multimedia, localización basada en GPS, etc.

- **Filosofía de dispositivo siempre conectado a Internet.** Está diseñado para la filosofía actual de "conectividad total", en la que el usuario sea capaz desde su dispositivo de acceder a cualquier dato u información.

- **Aceptable nivel de seguridad.** Aunque es quizá uno de los puntos más débiles del sistema operativo, el nivel de seguridad es aceptable, gracias entre otras cosas a que cada aplicación dispone de una serie de permisos que limitan su rango de actuación.

- **Optimizado para baja potencia y poca memoria.** Utiliza la Máquina Virtual Dalvik, que se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.

- **Alta calidad de gráficos y sonido.** Trabaja con gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora codecs estándar más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC….

### 2.3.5 Arquitectura

Tal y como muestra la Figura 31, el Sistema Android está dividido en capas, desde un nivel más bajo de procesamiento (donde se incluye el Kernel), a un nivel más alto de procesamiento (donde se incluyen las aplicaciones), estas diferentes capas interactúan entre sí para llevar a cabo el correcto funcionamiento del sistema.

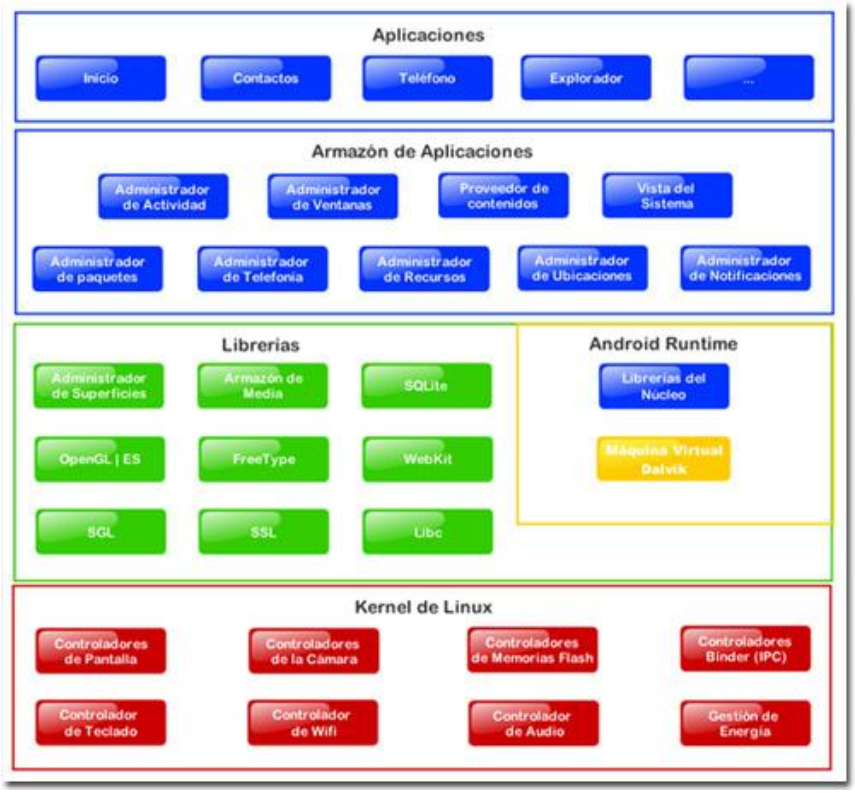



Figura 31: Módulos de un sistema de conversión de texto en habla

### Kernel

El núcleo del sistema operativo está basado en el mismo Kernel de Linux en su versión 2.6 y es muy similar al que se puede encontrar en cualquiera de las distribuciones de Linux.

El núcleo es el encargado de interactuar entre el hardware y el resto de capas de la arquitectura. De forma que esa es la única capa dependiente del hardware.

### Librerías

Esta capa queda situada justo por encima del Kernel. Dichas librerías están escritas en C o C++ y se compilan dependiendo de la arquitectura hardware específico del dispositivo correspondiente. Suelen estar hechas por el fabricante del dispositivo que a su vez se encarga de instalarlas en el mismo. Su objetivo es el de proporcionar funcionalidad a las aplicaciones para tareas que se repiten de forma frecuente. Así se evita tener que codificar estas tareas cada vez y el sistema funciona de forma más eficiente.

Las librerías más importantes del sistema operativo y que se utilizan actualmente para el desarrollo de la mayoría de aplicaciones son:

- **Activity Manager**. Se encarga de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.

- **Windows Manager.** Se encarga de organizar lo que se mostrará en pantalla. Básicamente crea las superficies en la pantalla que posteriormente pasarán a ser ocupadas por las actividades.

- **Content Provider**. Esta librería es muy interesante porque crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.

- **Views**. En Android, las vistas los elementos que nos ayudarán a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de Google Maps.

- **Notification Manager**. Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Un dato importante es que esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.

- **Package Manager**. Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con paquete nos referimos a la forma en que se distribuyen las aplicaciones Android, estos contienen el archivo .apk, que a su vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.

- **Telephony Manager.** Con esta librería podremos realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.

- **Resource Manager**. Con esta librería podremos gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o layouts. En un post relacionado a la estructura de un proyecto Android veremos esto más a fondo.

- **Location Manager.** Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.

- **Sensor Manager**. Nos permite manipular los elementos de hardware del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.

- **Cámara**: Con esta librería podemos hacer uso de la(s) cámara(s) del dispositivo para tomar fotografías o para grabar vídeo.

- **Multimedia.** Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

**Entorno de ejecución (Runtime)**

Su componente principal es la máquina virtual Dalvik. Debido a las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar, lo que llevo a Google a crear Dalvid para responder mejor a dichas limitaciones.

Una de sus ventajas es que las aplicaciones una vez compiladas en el formato especifico de Dalvik pueden ser ejecutadas en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación.

**Framework de aplicaciones**

La siguiente capa está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik.

Los servicios más importantes de esta capa son:

- **Views**: Parte visual de los componentes.

- **Resource Manager**. Proporciona acceso a recursos que son de código.

- **Activity Manager**. Maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.

- **Notificacion Manager**. Permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

- **Content Providers**: Mecanismo sencillo para acceder a datos de otras aplicaciones.

**Aplicaciones**

Esta es la última capa donde se incluyen todas las aplicaciones, tanto si tienen interfaz de usuario como las que no. En esta capa están también las aplicaciones nativas (programadas en C o C++), y las de administración (programadas en Java), además de todas aquellas que instale el usuario.

Una de las grandes ventajas de Android, como podemos ver, tras repasar su arquitectura es que nos proporciona un entorno sumamente poderoso para que podamos programar aplicaciones que hagan cualquier cosa. Nada dentro de Android es inaccesible y podemos jugar siempre con las aplicaciones de nuestro teléfono para optimizar cualquier tarea.

El potencial de Android se sitúa en el control total que se le da al usuario para que haga de su teléfono un dispositivo a su medida [ANG].

### 2.3.6 Ventajas y desventajas de Android

Como la utilización de cualquier Sistema Operativo, o cualquier sistema en el que tenemos varias opciones para elegir, siempre le podemos encontrar una serie de ventajas y desventajas en su uso, así como la comparación entre elementos, para saber cual es mejor antes de elegir uno de ellos.

En la red hay muchos blogs y páginas en las que podemos encontrar opiniones, y análisis sobre el sistema operativo presentando las características del mismo, por ejemplo en [SEB12]. Sin embargo de todos estos análisis y de la propia experiencia personal se pueden sacar una serie de conclusiones acerca de este sistema operativo.

**Ventajas**

Las ventajas son todos aquellos elementos y características que hacen del sistema operativo, el modelo a tener en cuenta a la hora de utilizarlo como nuestro sistema de trabajo para nuestro dispositivo. Algunas de estas características positivas en Android son:

- La más importante y principal, aquella que los expertos, así como casi totalidad de los usuarios de Android no dudan en destacar por encima de las demás se trata de las posibilidades que tiene Android por ser un sistema operativo de código abierto. De forma que se existe mucha mayor flexibilidad e integrabilidad de todas sus aplicaciones.
- Otra ventaja, probablemente derivada de la anterior, es que existen en el mercado actual un alto número de aplicaciones, para casi todos los ámbitos y orientadas a casi cualquier aspecto.
- Android además es un sistema operativo totalmente multitarea, es decir, se pueden tener abiertas varias aplicaciones a la vez para su uso (por ejemplo navegar mientras se escucha música), sin embargo hay que tener cuidado con esto, puesto que el tener muchas aplicaciones a la vez puede suponer que el dispositivo funcione más lentamente.

- Al ser un sistema operativo desarrollado por Google tiene total integrabilidad con todos sus elementos (Gmail, Google Maps, iGoogle, Calendar, etc.), siendo muchas de estas herramientas en la actualidad indispensables para muchos usuarios de la red, y a través de este sistema operativo móvil se pueden usar en cualquier lugar y cualquier momento de forma más rápida y eficiente.
- El gran nivel de personalización que tienen los dispositivos Android. La inmensa facilidad que presentar para darle un toque personal a cada una de las pantallas, iconos, interfaces y todas las aplicaciones que se pueden utilizar este sistema operativo hacen que sea imposible encontrar dos dispositivos móviles con las mismas características de personalización.

**Desventajas**

Las desventajas son aquellos elementos que nos hacen plantearnos si realmente merece la pena utilizar este sistema para nuestro dispositivo móvil. Y aunque parece una paradoja, en el caso de Android, muchas de sus desventajas vienen ligadas a sus mayores ventajas:

- La principal desventaja que tienen los dispositivos que utilizan Android viene debido al uso de la batería. En la mayor parte de los casos, la posibilidad de tener funcionando a la vez varias aplicaciones (la ventaja de la multitarea), así como la alta usabilidad que se da de muchas de ellas, hace que el dispositivo este continuamente trabajando y consumiendo recursos, lo que hace que el nivel de batería disminuya de forma considerable.
- El hecho de que sea código abierto permite con mayor facilidad la existencia de aplicaciones malware o software malintencionado, que dañe nuestro dispositivo móvil.
- Puede resultar complicado, engorroso y poco intuitivo. Debido a todas las opciones de personalización que permite, y la cantidad de elementos que existen para la modificación de interfaz, aspecto y manejo del mismo.

### 2.3.7 Comparativa con otros sistemas operativos móviles

Y aunque las comparaciones son odiosas, la forma más fácil de conocer las ventajas y desventajas de un sistema radica en compararlo con otros sistemas que realicen las mismas tareas o al menos muchas de ellas en un ámbito similar.

En esta comparativa nos centraremos en los sistemas operativos para dispositivos móviles más extendidos en la actualidad, a saber: Android, Apple iOS, BlackBerry, Windows Phone y Symbian (Figura 32).


Figura 32. Comparativa interfaces principales SO dispositivos móviles

Podemos empezar por la cuota de mercado que tienen cada uno de los sistemas operativos móviles. En la figura X podemos observar que desde el año 2009 donde el sistema operativo móvil más extendido era Symbian, con una cuota de mercado cercana al 45% del total mundial, el panorama ha cambiado muy notablemente.

Si bien es cierto que al inicio de la gráfica Symbian captaba un gran número del mercado mundial, probablemente debido a que de los sistemas de la comparativa era el más longevo de todos, teniendo en cuenta que la mayoría de los demás aún “estaban en pañales” y apenas acababan de dar sus primeros pasos, de forma que el número de dispositivos móviles que los montaban eran escasos.

Todo cambia a partir del año 2010/2011 cuando la explosión de los dispositivos *Smartphone* se convierte en un boom mundial. Esto hace que los dispositivos que montan Symbian comiencen a reducirse en número debido a que el fabricante Nokia, el propietario de dicho Sistema Operativo no es capaz de avanzar tan rápidamente a nivel hardware en sus dispositivos como sus más directos competidores, que montan otro tipo de sistemas, principalmente hablamos de Android y Apple iOS, que resultan ser pioneros en cuanto a los dispositivos se refiere.

Por otro lado, el sistema Blackberry que resulto ser muy novedoso en su momento comienza a perder también cuota de mercado, probablemente porque se asocia más a un ámbito profesional que a un ámbito personal debido a sus características y debido a que la mayoría de sus terminales continúan con la estructura de teclado físico junto con la pantalla que cada vez es menos demandado por los usuarios. Finalmente Windows Phone es un sistema que quizá comienza a asentarse “demasiado tarde”, cuando muchos de sus competidores tienen una amplia variedad de aplicaciones o terminales, Windows Phone queda muy atrás en este aspecto.

La gran diferenciación que existe actualmente entre iOS y Android frente a sus más directos competidores radica seguramente en primer lugar a que en el caso de Android aporta mayor control sobre la plataforma tanto para fabricantes como para operadores móviles. En el caso de iOS es diferente, ya que, al ser Apple también fabricante, únicamente tiene que lidiar con los operadores móviles que en la mayoría de los casos terminan pasando por el aro.

Sin embargo, tal y como muestra la Figura 33, vemos que actualmente Android copa prácticamente un porcentaje cercano al 80% de la cuota de mercado mundial, con un amplio margen de distancia frente a su más cercano competidor, iOS de Apple. Probablemente esta gran diferencia se pueda achacar a que el sistema de Apple únicamente se monta en los dispositivos de su propia compañía, estando muy limitado al número de terminales. Sin embargo Android es utilizado por una gran número de fabricantes de terminales móviles que montan en ellos este Sistema Operativo. Esto, asociado al carácter "elitista" que se le presupone a los terminales de la compañía Apple y que, ciertamente, suponen en la mayoría de los casos una gran diferencia de precio frente a los terminales que montan Android, sean los verdaderos motivos de la explosión del Sistema Operativo Open Source.

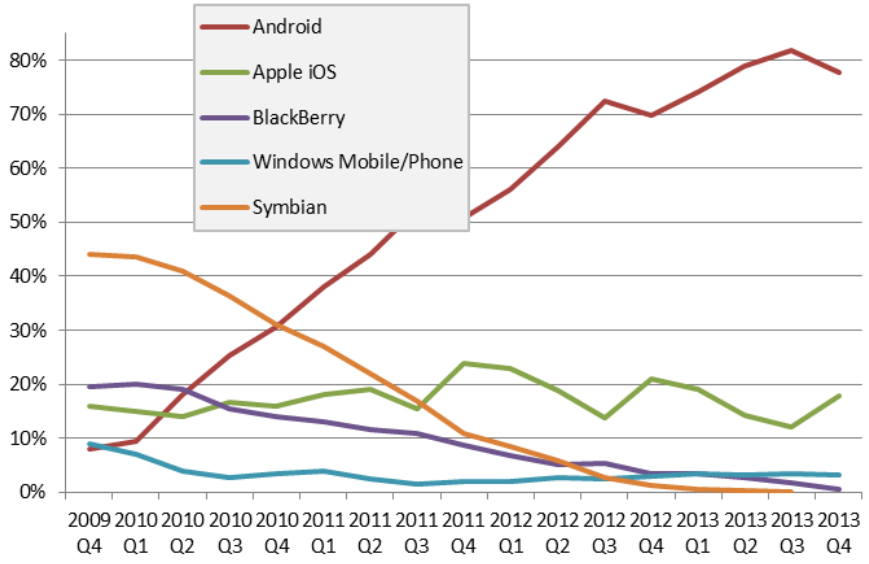



Figura 33. Cuota de mercado SO dispositivos móviles a lo largo de los años

En la Tabla 1 se puede observar una comparativa rápida y completa entre los sistemas operativos móviles de los que hablamos.

| Fabricante | Google | BlackBerry OS 10 | Apple | Microsoft |
|---|---|---|---|---|
| **Nombre** | Android 4.4 | BlackBerry OS 10 | iOS 6.1.2 | Windows Phone 8 |
| **Arquitecturas soportadas** | ARM, MIPS, x86 | ARM | ARM | ARM |
| **VPN** | Sí | No de serie | Sí | No |
| **Navegación GPS** | Sí | Sí | Sí | Sí |
| **Soporte de NFC** | Sí | Sí | No | Sí |
| **Dispositivos** | Smartphone y tabletas | Smartphone | Smartphone y tabletas | Smartphone |
| **Tipo de interfaz** | Icono y widgets | Baldosas e iconos | Iconos | Baldosas animadas |
| **Programado en** | C, C++ y Java | C y C++ | C, C++ y Objective-C | XNA, .NET, C#, C, C++ y VB.NET |
| **Control parental** | No de serie | Sí | Sí | Sí |
| **Multitarea** | Sí | Sí | Limitada | Sí |
| **Soporta idiomas adicionales al inglés** | Sí | Sí | Sí | Sí |
| **Gestor de paquetes** | APK | No disponible | iTunes | Zune Software |
| **Tethering** | Sí | Sí | Sí | Sí |
| **Notificaciones** | Sí | Sí | Sí | Sí |
| **Streaming multimedia** | No de serie | No | AirPlay | SmartGlass |
| **Bloqueo y eliminación de datos en remoto** | No de serie | No de serie | Sí | Sí |
| **Sincronización con la Nube** | Google Drive | No de serie | iCloud | SkyDrive |
| **Mapas** | Google Maps | BlackBerry Maps | Apple Maps | Nokia Maps, Bing Maps |
| **Reconocimiento de voz** | Sí | Sí | Sí | Sí |
| **Métodos de entrada alternativos** | Sí | No de serie | No de serie | No de serie |
| **Corrector ortográfico** | Sí | Sí | Sí | Sí |
| **Motor del navegador web** | WebKit | WebKit | WebKit | Trident |

| | | | | |
|---|---|---|---|---|
| **Servicio remoto de localización GPS** | No de serie | No de serie | Sí | Sí |
| **Pagos móviles** | Google Wallet | No de serie | Apple Passbook | Microsoft Wallet |
| **Soporta Flash** | Sí | Sí | No | No |
| **Asistente de voz** | Google Now | No de serie | Siri | No de serie |
| **Familia del sistema operativo** | Linux | Mobile OS | Darwin | Windows NT |
| **Tipo de licencia** | Libre | Propietaria | Propietaria | Propietaria |
| **VoIP / video chat** | Google Talk | No de serie | FaceTime | Skype |
| **Ofimática** | Google Docs | Docs to Go | No de serie | Office |
| **Mensajería** | Gchat / Gtalk | BBM | iMessage | Skype |
| **Integra Facebook** | No de serie | No de serie | Sí | Sí |
| **Integra Twitter** | No de serie | Sí | Sí | No de serie |
| **Cuenta de invitado** | No de serie | No | Sí | Sí |
| **Cifrado de datos** | Sí | Sí | Sí | Sí |
| **Tienda de apps** | Google Play | App World | App Store | Windows Phone Store |
| **Actualizaciones OTA (Over the Air)** | Sí | Sí | Sí | Sí |
| **Accesibilidad** | Sí | Sí | Sí | Sí |
| **Multiusuario** | Sí | No | No | No |

Tabla 1. Comparativa características principales SO dispositivos móviles

### 2.3.8 Entorno de desarrollo

Como se ha comentado con anterioridad, una de las principales, y probablemente la mejor ventaja que tiene Android es que se trata de software libre y es prácticamente en su totalidad de código abierto. Esto quiere decir que con un entorno de desarrollo adecuado y con un poco de conocimiento de programación Java cualquiera puede crear sus propias aplicaciones y luego utilizarlas en su propio sistema o dispositivo móvil.

Todo lo necesario para poder tener listo y disponible un entorno de desarrollo adecuado para construir nuestras propias aplicaciones Android es lo siguiente:

**Un IDE de desarrollo adecuado**

Sin olvidar que para realizar tareas de programación, únicamente es necesario tener un procesador de texto plano que nos permita escribir nuestro programa, es cierto que todo se hace mucho más fácil con la utilización de un IDE (Entorno de desarrollo integrado) [PET12]. Y en el caso de desarrollar aplicaciones Android el IDE más generalizado es Eclipse [JYC]. Esta herramienta facilita enormemente las tareas de edición, compilación y ejecución de programas en la fase de desarrollo de los mismos.

El por qué del éxito de eclipse en el desarrollo de aplicaciones Android, es debido a que aunque soporta varios lenguajes de integración, aquel con el que está mayormente integrado es con Java, lenguaje que le ha llevado a estar entre los IDEs más populares del mercado. Este hecho junto con que el propio lenguaje Java es el utilizado en el desarrollo Android han catapultado a Eclipse como uno de los IDEs más utilizados en la actualidad en todo el mundo.

Además Eclipse también tiene la ventaja de que se puede descargar de forma gratuita e instalar de forma rápida y sencilla en cualquiera de los sistemas operativos más actualizados en la actualidad: Windows, Linux y Solaris.

**El JDK de Java**

Aunque realmente el JDK (Java Development Kit) [TAR12], no es estrictamente necesario para compilar las aplicaciones que desarrollemos, sí que es necesario en el caso de utilizar nuestro IDE, y las cosas se hacen mucho más sencillas. El JDK es único y exclusivo para Java y nos permite que nuestras aplicaciones desarrolladas en Eclipse se compilen y depuren de forma rápida. Al igual que ocurre con Eclipse, el JDK es distribuido de forma gratuita por Oracle para los sistemas operativos habituales: Linux, Windows y Solaris.

**El SDK de Android**

El SDK (Kit de desarrollo Software) [PRO12], son una serie de herramientas orientadas al desarrollo de aplicaciones Android, integradas en un IDE. De forma que introduce las librerías, opciones de compilación, máquinas virtuales para depuración y otra serie de elementos útiles en nuestro IDE que nos permiten integrar de forma rápida todo lo necesario para realizar un desarrollo completo de la aplicación.

Actualmente Google siguiendo con sus políticas de facilitar a los usuarios cada vez más el acceso al desarrollo de nuevas aplicaciones, permite descargar desde la página oficial de desarrolladores de Android un completo paquete de Desarrollador para nuevos usuarios al que denominan ADT Bundle y que incluye los siguientes elementos [DEV14]:

- Eclipse + Android Developer Tools plugin

- Android SDK Tools
- Android Platform Tools
- Ultima versión de la plataforma Android
- Ultima versión de la imagen del Sistema Android para el Emulador que permite probar las aplicaciones desarrolladas.

### 2.3.9 Instalación de aplicaciones en Android

La cuestión es, si yo tengo un dispositivo Android, ¿Qué puedo hacer para empezar a instalar y a manejarme mediante aplicaciones?

Android debido a su gran extensión de aplicaciones, permite realizar muchas tareas con nuestro dispositivo móvil, pero la mayoría de estas tareas se realizan a través de aplicaciones que tienen que ser instaladas para poder usarse en el dispositivo. Los métodos para ellos son los siguientes:

**Mediante el PC a través de Google Play**

Para este método es necesario tener una cuenta de Google asociada al dispositivo. Una vez registrados y en posesión de una cuenta de google, únicamente hay que acceder a Google Play a través de la página principal de google, buscar la aplicación a través del buscador, seleccionar el dispositivo en el que queremos que se instale la propia aplicación y aceptar los términos. Un ejemplo de instalación se puede ver en la Figura 34.


Figura 34: Ejemplo de instalación de aplicación a través de Google Play

**Mediante el propio dispositivo con una conexión a internet a través del Market de Android**

Todos los dispositivos Android cuentan con una aplicación “interna”, que actualmente se llama *Play Store* y que suele estar disponible desde el propio escritorio del dispositivo. El funcionamiento es muy similar que el de Google Play, pero en vez de, desde el PC, la búsqueda e instalación se realizan desde el propio dispositivo, que requiere obviamente conexión a Internet para realizar esta tarea.

**A través del dispositivo sin necesidad de conexión a Internet mediante elementos externos**

En el caso de no disponer de conexión a internet, existe la posibilidad de introducir aplicaciones mediante elementos externos de almacenamiento, como por ejemplo, una microSD. Estos archivos de instalación tendrán extensión .apk y será necesario realizar una serie de pasos “extras” para que el dispositivo permita su instalación:

Para poder instalarlas en nuestro teléfono debemos activar lo siguiente:

**Ajustes –> Aplicaciones –> Orígenes desconocidos**

Posteriormente habrá que utilizar un programa explorador de archivos, tal como WinSCP o bien, para poder instalarla o simplemente buscamos la aplicación y pulsamos para instalarla. Todas estas aplicaciones, al no estar instaladas desde el Market, se tienen que desinstalar desde:

**Ajustes –> Aplicaciones –> Administrar aplicaciones**

## 2.3.10 Reconocimiento de voz y Android

El reconocimiento de voz en Android viene incluido de forma nativa desde la primera versión del sistema operativo, sin embargo, aunque actualmente es una de las herramientas con más éxito del sistema, inicialmente no tenía tanta potencia, por lo que se desarrollaron multitud de aplicaciones externas que realizaban esta función.

Analizamos algunas de las alternativas gratuitas en Android:

- **Speaktoit.** Es probablemente, en la actualidad la aplicación por voz más utilizada del Market de Android. Se trata de un asistente personal para realizar más fácilmente todo tipo de tareas: Puede resolver dudas, realiza tareas en tu Smartphone por ti, o puede planificar notificaciones para que te avise frente a eventos que te interesen. Pero, principalmente la mayor novedad que tiene este asistente es que no necesita conocer unos

comandos de voz específicos, si no que para comunicarse con el asistente puede utilizarse un lenguaje neutro y natural.

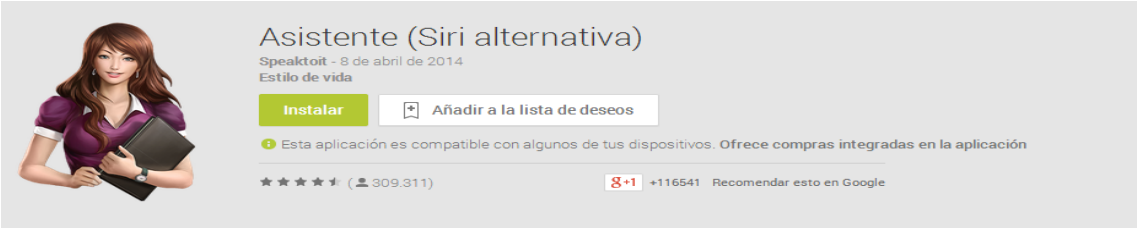



Figura 35. Aplicación Asistente Android

- **Skyvi.** Es otro tipo de asistente que funciona mediante el reconocimiento de voz. Permite operaciones tales como escribir texto, buscar y almacenar direcciones, realizar llamadas, consultar las condiciones meteorológicas. Es especialmente acertada para consulta de direcciones y para los recordatorios, ya que con otro tipo de preguntas al azar tiene mayores dificultades.

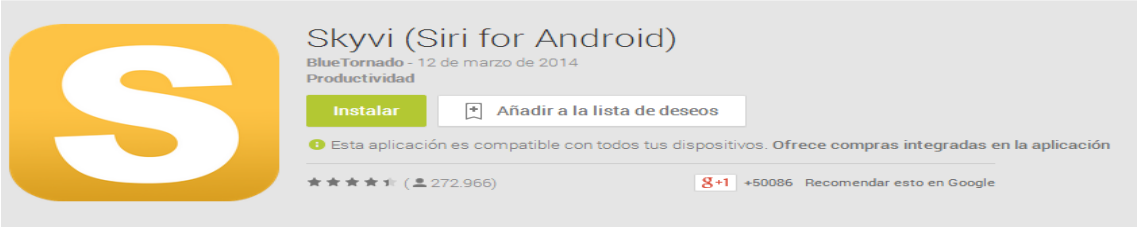



Figura 36. Aplicación Skyvi Android

- **Sherpa.** También al estilo de asistente personal, esta aplicación tiene una serie de comandos de voz concretas para realizar las acciones que desea el usuario. Su particularidad radica en que en sus últimas versiones es capaz de “aprender” de los gustos del usuario. Debido a la repetición de ciertos patrones de comandos por voz realizados por el usuario es capaz de anticiparse a futuras peticiones bajo ciertas condiciones.


Figura 37. Aplicación Sherpa Android

Como se puede ver, después de describir algunas de las aplicaciones más exitosas en el Market *Play Store* de Android y que utilizan el reconocimiento

de voz para trabajar, en su gran mayoría están orientadas a trabajar como si de una especie de asistente personal se tratase.

Aunque el mercado de aplicaciones en Android es muy grande, y hay aplicaciones de todo tipo, y en concreto que utilizan el reconocimiento de Voz como entrada de datos, otro tipo de aplicaciones que no estén orientadas a asistentes personales en este ámbito son poco populares actualmente en el mercado.

Aún así los propios fabricantes van incluyendo cada vez más aplicaciones nativas en los dispositivos que se utilizan mediante comandos de voz:

- El propio asistente nativo de Android, como es *Google Now.*
- Aplicaciones de búsqueda de información en internet.
- Aplicaciones de mapas.
- Cámara de los dispositivos activada mediante comandos de voz.

En la comparativa entre el sistema de reconocimiento de voz nativo de Android y el reconocimiento de voz de iOS, la empresa Fortune realizó recientemente un estudio de 1600 preguntas idénticas a ambos servicios, 800 de ellos en un ambiente cerrado y otras 800 en una calle de Minneapolis para probar la tolerancia al ruido. Los resultados han sido los siguientes

- Google **comprende el 100%** de las preguntas. Tanto en ambiente cerrado como en ambiente “hostil”.
- Google **responde bien el 86%** de las veces
- Siri comprende el **83%** de las peticiones en un ambiente ruidoso, el **89%** en silencio
- Siri responde correctamente el **62%** en la calle y el **68%** en una sala silenciosa Además tiene el hándicap de que el 60% de las búsquedas a las respuestas satisfactorias llegan desde el buscador de la competencia, es decir, desde Google.

Una comparativa más detallada mediante una comparación visual, en vídeo se puede ver en [XAT12].

### 2.3.11 Aplicaciones de síntesis de texto a voz en Android

En Android se incorpora un motor TTS instalado por defecto, con el cual las aplicaciones (o el propio sistema) pueden traducir textos en una señal de voz. Sin embargo, Android nos da la posibilidad de instalar y personalizar varios motores adicionales [GAB11]:

- **SVOX:** Actualmente es una de las pocas opciones de calidad que aparecen como alternativa a la propia síntesis de voz que trae de serie Android. Por desgracia eso también significa que no se trata de una aplicación gratuita. Sin embargo en su versión de prueba de dos semanas se pueden encontrar hasta 25 idiomas diferentes con 40 voces en total en género femenino o masculino.

  Una de las desventajas que tiene es que la voz suena poco natural, además de que para que funcione es necesario que se instale la aplicación gratuita "Classic Text To Speech Engine" desde *Google Play* además de la voz elegida en el idioma correspondiente.

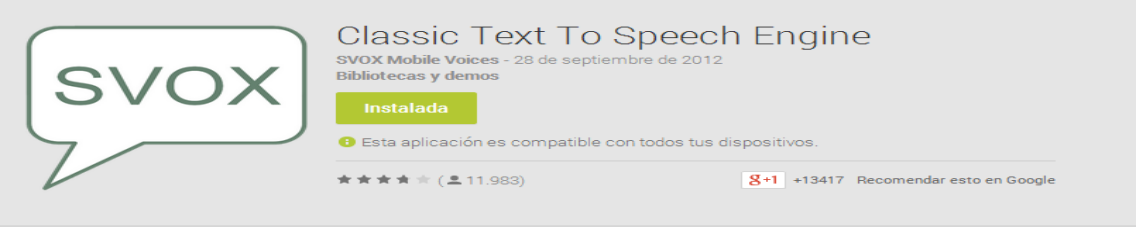



Figura 38. Aplicación SVox Android

- **IVONA:** Aunque aún se trata de una aplicación en fase beta, se trata de una de las aplicaciones más completas que hay en la actualidad para la síntesis de voz en Android. El hecho de no ser una versión definitiva supone que, al menos de momento, es una aplicación *gratuita*.

  Presenta un total de 13 voces, y su principal ventaja es que las voces tienen un sonido muy natural frente a otras opciones que presentan una opción de estilo robótico a la que podemos estar acostumbrados en muchas aplicaciones de de este estilo. Además de que utiliza el motor BrightVoice para ofrecer una lectura más expresiva y sin apenas cortes.

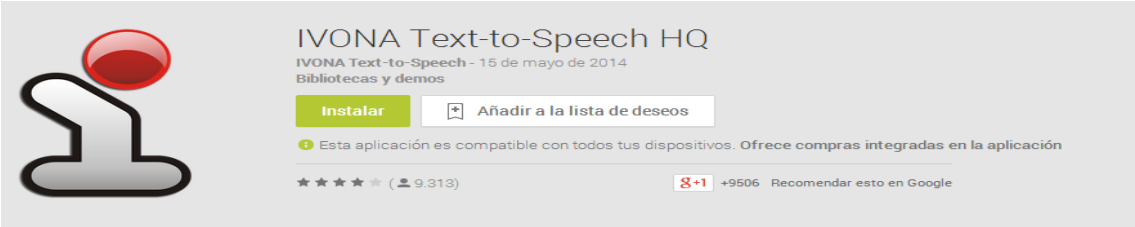



Figura 39. Aplicación *Ivona* Android

- **SÍNTESIS DE VOZ DE GOOGLE:** La aplicación de Síntesis de Voz de Google para Android, viene de serie actualmente en muchos dispositivos. Sin embargo algunos fabricantes aún continúan incluyendo sus propios motores de síntesis de voz (TTS). Esta aplicación se encuentra disponible a partir de la versión de Android 4.0.3 o superior e incluye los idiomas:

alemán, coreano, español, francés, inglés (EEUU y Reino Unido) e italiano.

Actualmente se puede poner como predeterminado realizando los siguientes pasos: Ajustes > Idioma e introducción de texto > Síntesis de voz y seleccionando el moto de síntesis de voz de Google.

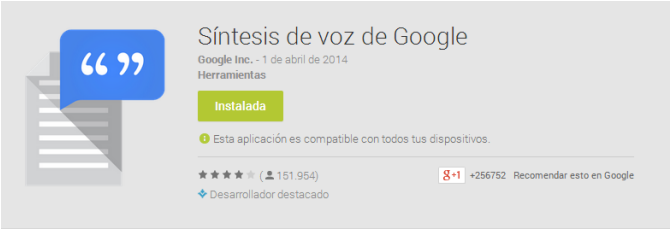



**Figura 40. Aplicación de *Síntesis de Voz de Google* para Android**

# Capítulo 3. Descripción de la aplicación y de los módulos del sistema

En este capítulo se realizará una descripción detallada de las características principales de las que consta la aplicación realizada, en lo que se refiere a su funcionalidad, decisiones de diseño, arquitectura y diagrama de bloques que sigue la estructura de la misma.

## 3.1 Arquitectura del sistema

La aplicación está diseñada para facilitar la tarea a un oficial de un equipo, en concreto orientado al deporte del balonmano, que realiza las estadísticas en tiempo real durante un partido. Esta adaptada para poder realizar las funciones tanto a través de comandos de voz como mediante la pantalla. De forma que posteriormente estos datos quedan almacenados en el dispositivo utilizado y podrán ser consultados.

La aplicación desarrollada ofrece al usuario la posibilidad de realizar las siguientes tareas:

- Generación de estadísticas en tiempo real.
- Consulta de resultados de partidos realizados en la aplicación.
- Consulta de estadísticas totales de jugadores y equipos de los partidos realizados en la aplicación.
- Gestión de equipos y jugadores.

En la Figura 41 se puede ver una representación gráfica de las funcionalidades generales ofrecidas por el sistema.


Figura 41. Arquitectura del sistema

La aplicación está basada principalmente en la utilización de los comandos de voz para el manejo de la aplicación como alternativa al uso habitual del modo táctil. En la figura 42 se puede ver un diagrama de funcionamiento de los reconocimientos de voz del sistema.

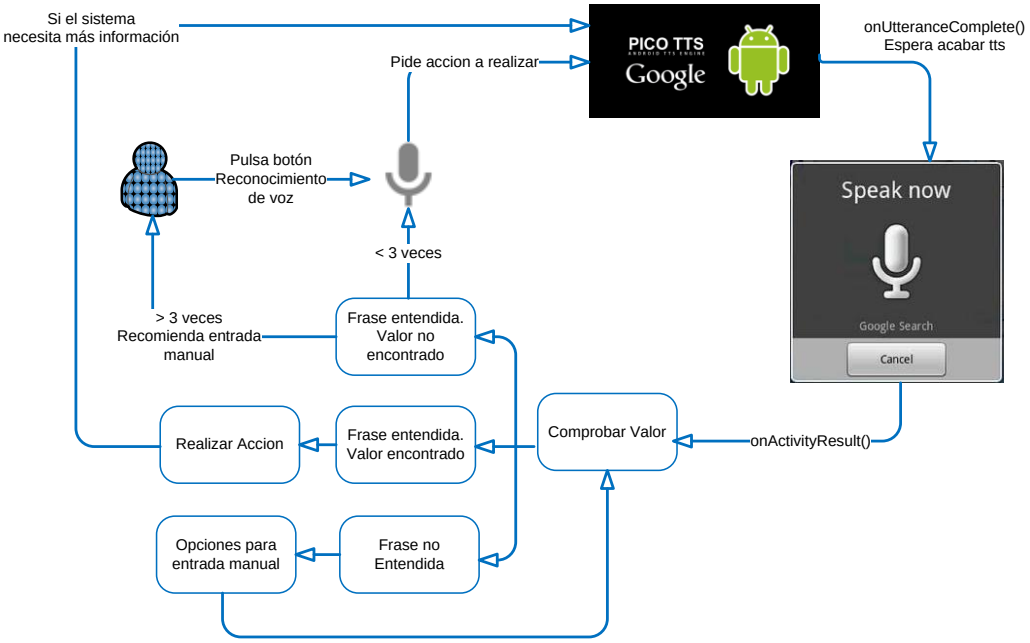



Figura 42. Diagrama para los Reconocimientos de voz de la aplicación

# 3.2 Diseño de la BBDD

A continuación se describe la estructura de la BBDD que se ha utilizado para gestionar la aplicación y que se guardará de forma interna en el dispositivo del usuario para poder almacenar, consultar y manejar la información necesaria.

Se divide la aplicación en dos tipos de BBDD:

- en primer lugar la **BBDD de Temporada**, la cual será única para gestionar todos los partidos de los equipos, en esta BBDD se guardará la información de los resultados de todos los equipos de los que tenemos datos además de las estadísticas acumuladas de todos los jugadores y porteros de los equipos. Se decide utilizar dos tablas diferentes, porteros y jugadores, pese a que ambas tendrán los mismo campos debido al carácter excepcional de esta posición puesto que a la hora de gestionar y mostrar la información se tratará de forma distinta y esto agilizará el trato de dicha información.

- en segundo lugar se creara una **BBDD de Partido** para cada enfrentamiento entre dos equipos. De forma que tendremos tantas BBDD de este tipo como partidos se hayan gestionado por la aplicación. El motivo será que es necesario gestionar la información de cada partido de forma individualizada para poder mostrarla de forma más rápida y así no interferir además entre datos de otros partidos.

En las Figuras 43 y 44 se muestra la estructura de cada una de los dos tipos de BBDD de la aplicación.

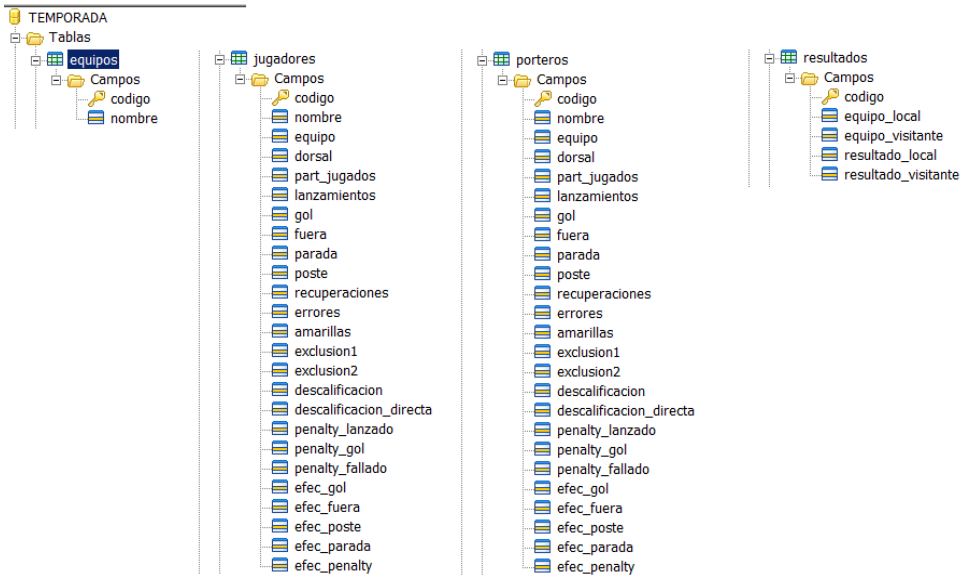



Figura 43. Estructura de la BBDD Temporada

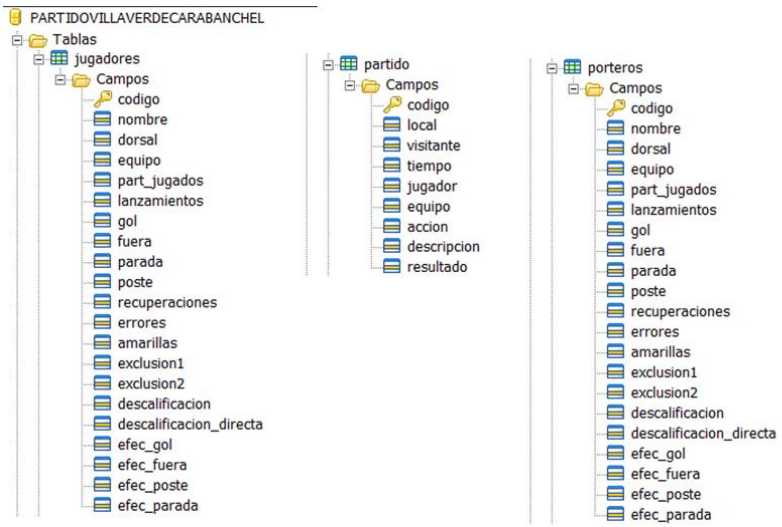



Figura 44. Estructura de la BBDD Partido

En las Tablas 2 y 3 se muestra el tipo de datos utilizados para cada uno de los campos de las BBDD.

| BBDD TEMPORADA | | | |
|---|---|---|---|
| **Tabla Equipos** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Nombre | String | | |
| **Tabla Jugadores** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Nombre | String | Equipo | String |
| Dorsal | Integer | Partidos Jugados | Integer |
| Lanzamientos | Integer | Goles | Integer |
| Fuera | Integer | Parada | Integer |
| Poste | Integer | Recuperaciones | Integer |
| Errores | Integer | Amarillas | Integer |
| Exclusiones | Integer | Descalificación | Integer |
| Descalificación Directa | Integer | Efectividad gol | Integer |
| Efectividad fuera | Integer | Efectividad poste | Integer |
| Efectividad parada | Integer | | |
| **Tabla Porteros** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Nombre | String | Equipo | String |
| Dorsal | Integer | Partidos Jugados | Integer |
| Lanzamientos | Integer | Goles | Integer |
| Fuera | Integer | Parada | Integer |
| Poste | Integer | Recuperaciones | Integer |
| Errores | Integer | Amarillas | Integer |
| Exclusiones | Integer | Descalificación | Integer |
| Descalificación Directa | Integer | Efectividad gol | Integer |
| Efectividad fuera | Integer | Efectividad poste | Integer |
| Efectividad parada | Integer | | |
| **Tabla Resultados** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| equipo_local | String | Equipo_visitante | String |
| Resultado_local | Integer | Resultado_visitante | Integer |

Tabla 2. Comandos de voz para la interfaz *Menú Principal*

| BBDD PARTIDO | | | |
|---|---|---|---|
| **Tabla Partido** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Local | String | Visitante | String |
| Tiempo | Time | Jugador | String |
| Equipo | String | Acción | String |
| Descripción | String | Resultado | String |
| **Tabla Jugadores** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Nombre | String | Equipo | String |
| Dorsal | Integer | Partidos Jugados | Integer |
| Lanzamientos | Integer | Goles | Integer |
| Fuera | Integer | Parada | Integer |
| Poste | Integer | Recuperaciones | Integer |
| Errores | Integer | Amarillas | Integer |
| Exclusiones | Integer | Descalificación | Integer |
| Descalificación Directa | Integer | Efectividad gol | Integer |
| Efectividad fuera | Integer | Efectividad poste | Integer |
| Efectividad parada | Integer | | |
| **Tabla Porteros** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| Nombre | String | Equipo | String |
| Dorsal | Integer | Partidos Jugados | Integer |
| Lanzamientos | Integer | Goles | Integer |
| Fuera | Integer | Parada | Integer |
| Poste | Integer | Recuperaciones | Integer |
| Errores | Integer | Amarillas | Integer |
| Exclusiones | Integer | Descalificación | Integer |
| Descalificación Directa | Integer | Efectividad gol | Integer |
| Efectividad fuera | Integer | Efectividad poste | Integer |
| Efectividad parada | Integer | | |
| **Tabla Resultados** | | | |
| **Campo** | **Tipo** | **Campo** | **Tipo** |
| equipo_local | String | Equipo_visitante | String |
| Resultado_local | Integer | Resultado_visitante | Integer |

Tabla 3. Comandos de voz para la interfaz *Menú Principal*

# 3.3 Interfaz Menú Principal

## 3.3.1 Funcionalidad

El interfaz ***Menú Principal*** implementa la primera pantalla que se le presenta al usuario al abrir la aplicación. A partir de este interfaz inicial la aplicación se subdivide en otros 4 interfaces secundarios entre los que el usuario puede elegir, los cuales tienen funcionalidades totalmente diferentes en función de la actividad que quiera realizar: Nuevo Partido, Resultados, Estadísticas y Editar Equipos. Por lo tanto este interfaz inicial es el que hará de hilo conductor de los demás.

Además, desde esta primera interfaz se pueden consultar los datos de reconocimientos que se han realizado en la aplicación desde la última vez que se ha abierto. De forma que se puede conocer si los reconocimientos realizados

han sido en su mayoría correctos, si el entorno en los que se han realizado es el adecuado o si por el contrario el número de fallos ha sido tan elevado que no se recomienda utilizar esta situación de reconocimiento para futuras actuaciones de la aplicación. En la figura 45 se puede ver una imagen de esta interfaz.


Figura 45. Interfaz del Menú Principal

## 3.3.2 Arquitectura

- **Estructura UML**

Esta interfaz principal está compuesta por el archivo MenuPrincipal.java, el cual tiene la estructura mostrada en la Figura 46.

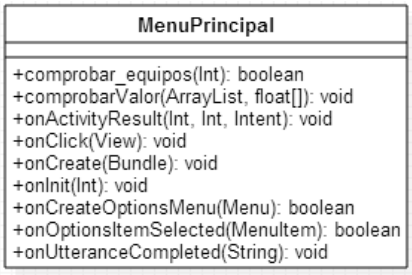



Figura 46. Diagrama UML de la clase MenuPrincipal.java

- **Estructura de la ActionBar**

En este caso la interfaz del ***Menú Principal*** estará compuesta por una ActionBar que contiene 3 iconos cuya funcionalidad se explica en la Figura 47.

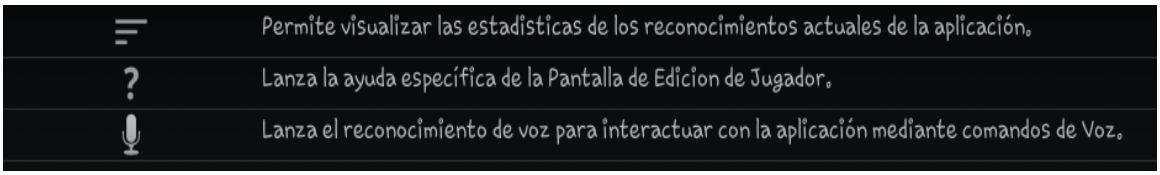



Figura 47. Botones de la ActionBar del Menú Principal

- **Comprobar estado de la BBDD**

Se hace necesario realizar la comprobación del estado de la BBDD de la aplicación para conocer el número de equipos que existen actualmente en la misma. Esto es debido a que de las 4 opciones permitidas a realizar desde este

menú principal: ***Nuevo Partido, Estadísticas, Resultados, Editar Equipo***; para las 3 primeras es necesario que existan al menos 2 equipos en la BBDD. Esto es debido a que para comenzar un nuevo partido son necesarios como resulta evidente 2 equipos, y para el caso de las estadísticas y los resultados, es necesario que haya finalizado 1 partido para que existan datos. Para realizar esta comprobación existe una rutina de nombre *comprobar_equipos* (Figura 48).

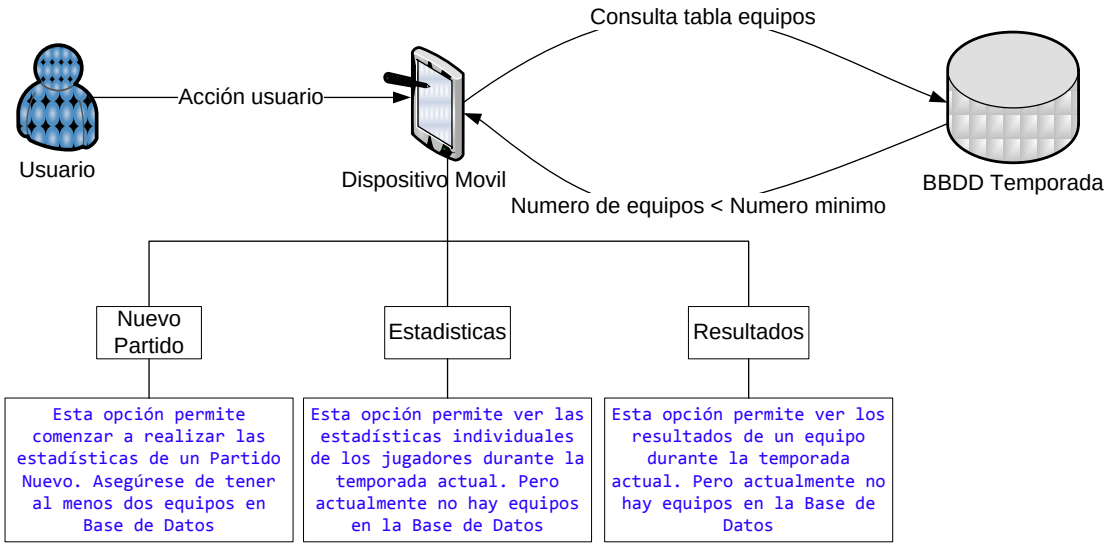



Figura 48. Diagrama de la función *comprobar_equipos*

## 3.3.3 Comandos de Reconocimiento de Voz

Los valores que son reconocidos mediante el Reconocimiento de Voz de Android para esta interfaz son los siguientes:

| Comando de Voz | Descripción |
|---|---|
| Nuevo | Permite comenzar un nuevo partido seleccionando los equipos participantes. |
| Resultados | Permite visualizar las estadísticas de un equipo concreto durante una temporada. |
| Editar | Permite visualizar los resultados de un equipo concreto durante una temporada. |
| Estadísticas | Permite editar los jugadores y equipos de la Base de datos de la aplicación. |

Tabla 4. Comandos de voz para la interfaz *Menú Principal*

## 3.3.4 Diagrama de flujo de la interfaz

En la Figura 49 se muestra el diagrama de flujo de esta interfaz, tanto para un usuario que elige la opción de actuación por voz como para un usuario que utiliza la opción de actuación táctil.

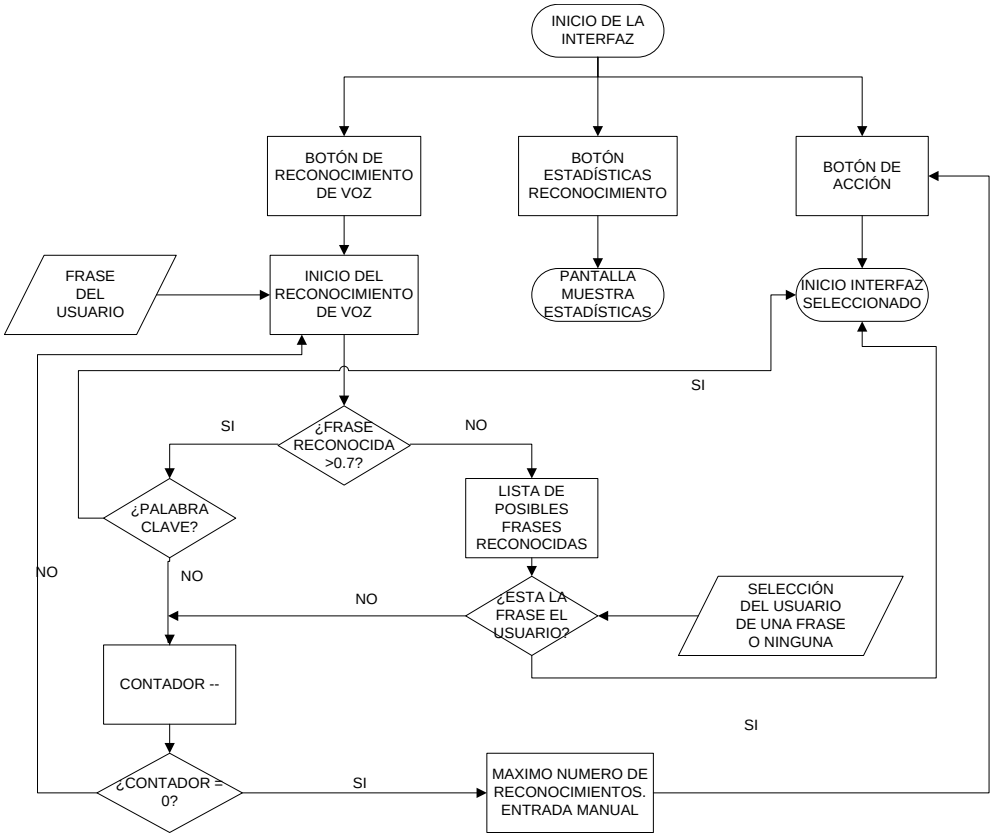



Figura 49. Diagrama de flujo de la interfaz Menú Principal

### 3.3.5 Escenario de uso

En la Figura 50 se muestra cómo se desarrolla un diálogo estándar para esta interfaz. “S” indica el diálogo correspondiente al sistema y “U” el diálogo correspondiente al usuario.

| | |
|---|---|
| S: | **¿Qué acción desea realizar**? |
| U: | Editar Equipo |
| S: | **Entrando en pantalla de Editar de Equipos** |

Figura 50. Escenario de Uso interfaz Menú Principal

## 3.4 Interfaz de Edición de Equipos

La interfaz de ***Edición de equipos*** será la que nos permite controlar de forma general la BBDD de nuestra aplicación, en lo que se refiere a los datos de la temporada para poder comenzar a realizar las estadísticas de los partidos. Los datos a establecer y las relaciones entre ellos los podemos ver en la Figura 51.

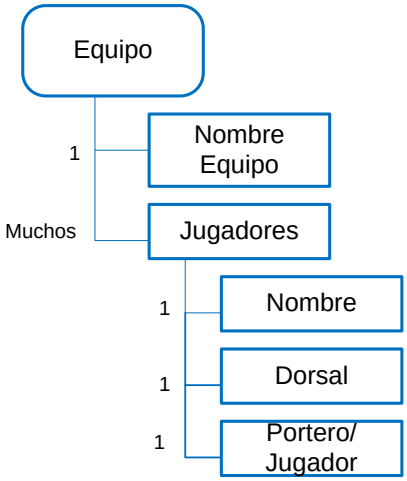



Figura 51. Relación de los datos de los equipos en la BBDD

En la Figura 52 podemos ver el Diagrama de las posibilidades que ofrece la línea de Edición de Equipos.

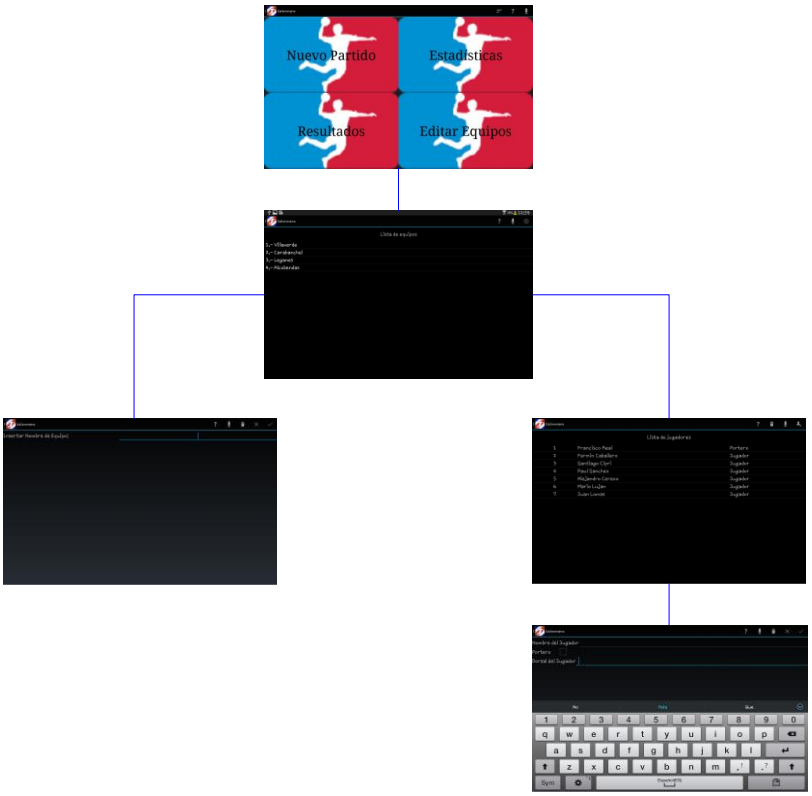



Figura 52. Diagrama de interfaces *EditarEquipos*

# 3.5 Sub-interfaz de Equipos

## 3.5.1 Funcionalidad

La interfaz ***Equipos*** se trata de una interfaz de tránsito entre las demás subinterfaces cuya finalidad es la de visualizar el nombre de todos los equipos

existentes en la Base de Datos, así como elegir si insertar uno nuevo o editar uno ya existente. En la Figura 53 se puede ver una imagen de esta interfaz.

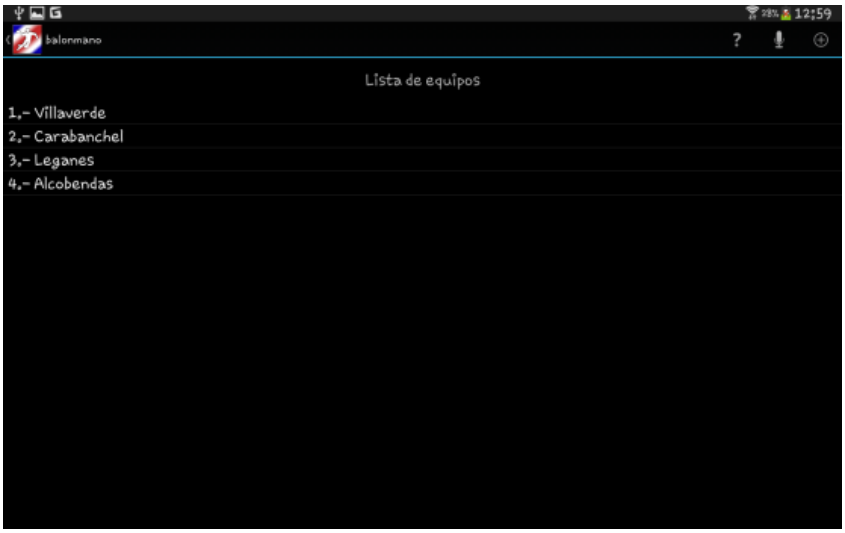



Figura 53. Interfaz de Nuevo Equipo

## 3.5.2 Arquitectura

- **Estructura UML**

Esta interfaz de inserción de equipo está compuesta por el archivo Equipos.java, el cual tiene la siguiente estructura:

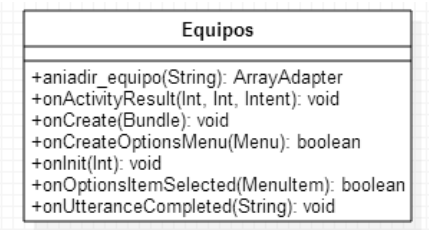



Figura 54. Diagrama UML de la clase Equipos.java

- **Estructura de la ActionBar**

En este caso la interfaz de ***Equipos*** estará compuesta por una ActionBar que contiene por 4 iconos cuya funcionalidad se explica en la Figura 55.

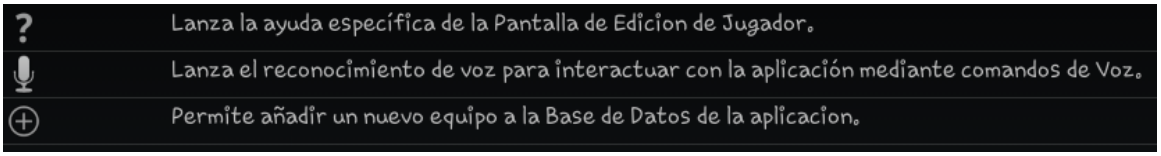



Figura 55. ActionBar de la interfaz Equipos

## 3.5.3 Comandos de Reconocimiento de Voz

Los valores que son reconocidos mediante el Reconocimiento de Voz de Android para esta interfaz son los siguientes:

| Comando de Voz | Descripción |
|---|---|
| Insertar/Añadir/Nuevo | Permite acceder a la subinterfaz para añadir un nuevo equipo en la Base de Datos de la aplicación. |
| Editar | Permite acceder a la subinterfaz que se utilizará para modificar un equipo existente en la Base de Datos. |

Tabla 5. Comandos de voz para la interfaz *Menú Principal*

### 3.5.4 Escenarios de uso

En las Figuras 56 y 57 se muestran dos ejemplos de escenarios de uso de esta interfaz:

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Nuevo |
| **S:** | **Ya puede iniciar las acciones para insertar un nuevo equipo en la Base de Datos** |

Figura 56. Escenario de uso de Inserción de Nuevo Equipo

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar**? |
| U: | Editar Equipo |
| **S:** | **Indique la posición del equipo a editar** |
| U: | 2 |
| **S:** | **Se cargarán los datos del equipo <Nombre_Equipo> para poder ser editados** |

Figura 57. Escenario de uso de Edición de Equipo

## 3.6 Sub-interfaz de Inserción de Equipos

### 3.6.1 Funcionalidad

La interfaz de ***Inserción*** permite añadir un equipo de la Base de Datos de la aplicación. La única funcionalidad de la misma es permitir al usuario una herramienta para introducir el nombre del Nuevo Equipo a introducir en la Base de Datos. En la Figura 58 se puede ver una imagen de esta interfaz.


Figura 58. Interfaz de *Inserción de Equipo*

## 3.6.2 Arquitectura

- **Estructura UML**

Esta interfaz de inserción de equipo está compuesta por el archivo Inserccion.java, el cual tiene la siguiente estructura:

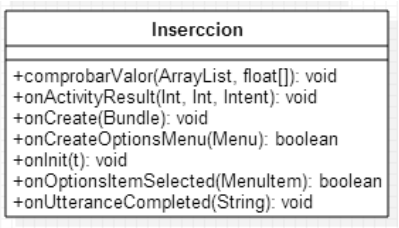



Figura 59. Diagrama UML de la clase Inserccion.java

- **Estructura de la ActionBar**

En este caso la interfaz de la interfaz ***Inserción*** estará compuesta por una ActionBar que contiene 4 iconos cuya funcionalidad se explica en la Figura 60.

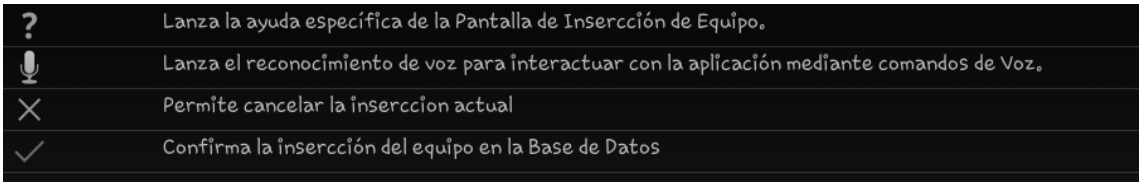



Figura 60. Botones de la ActionBar de la interfaz Inserción

- **Comprobaciones en la inserción**

Antes de realizar la inserción en la Base de Datos de la aplicación es necesario hacer una serie de comprobaciones para comprobar que el valor es correcto. Estas comprobaciones son las siguientes:

- El valor del campo de texto del nombre no está vacío.
- El nombre a insertar no existe previamente en la Base de Datos.
- Tiene una longitud adecuada al campo de la Base de Datos.

## 3.6.3 Comandos de Reconocimiento de Voz

Esta interfaz no tiene comandos específicos para el Reconocimiento de Voz. Únicamente se le piden datos de valor libre al usuario y/o confirmaciones.

## 3.6.4 Diagrama de flujo

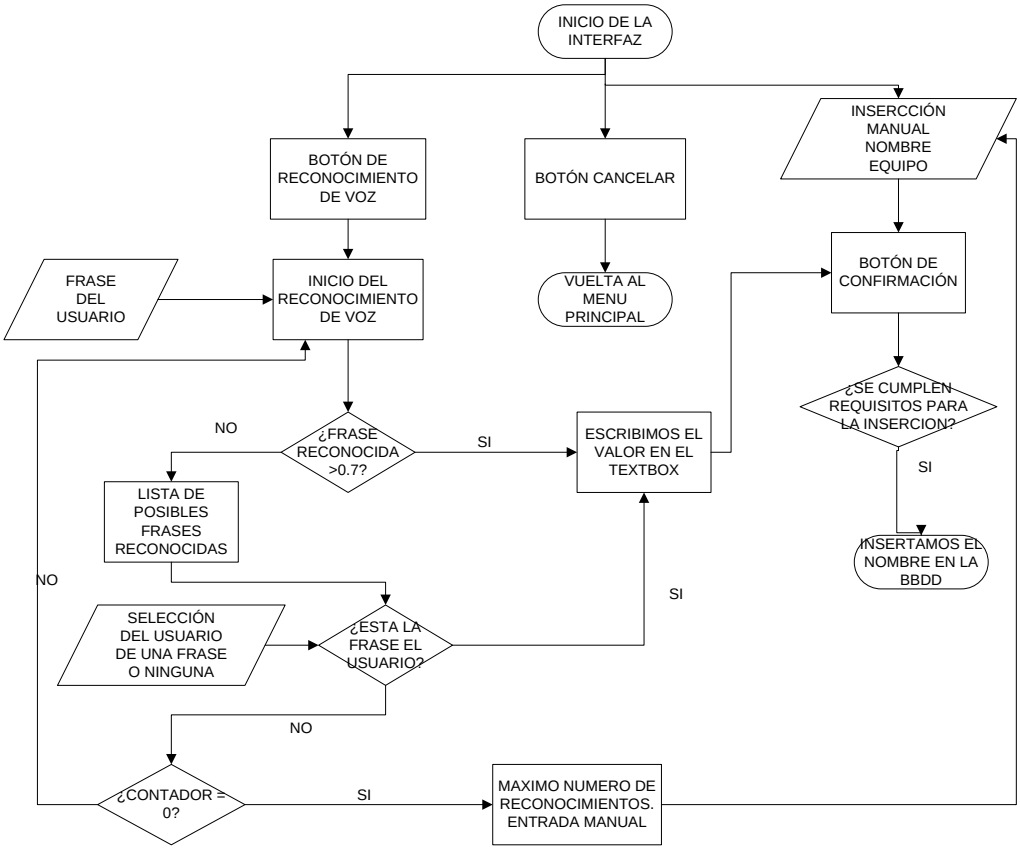



Figura 61. Diagrama de flujo de la interfaz Inserción

## 3.6.5 Escenarios de uso

En la Figura 62 se puede ver un ejemplo de diálogo entre el sistema y el usuario para la interfaz ***Inserción***. Dicho ejemplo representa las acciones necesarias para insertar de forma correcta un nuevo equipo en la Base de Datos de la aplicación mediante comandos de voz a través de esta interfaz.

| | |
|---|---|
| **S:** | **Indique el nuevo nombre para el equipo** |
| U: | Villaverde |
| **S:** | **¿Coincide el texto con el nombre del equipo deseado?** |
| U: | Si |
| **S:** | **Se ha insertado correctamente el equipo en la Base de Datos** |

Figura 62. Ejemplo de escenario de uso en la interfaz Inserción

# 3.7 Sub-interfaz de Jugadores del Equipo

## 3.7.1 Funcionalidad

Esta interfaz se utiliza para editar la información complementaria de un equipo en la Base de Datos de la aplicación. A través de ella podemos añadir jugadores a un equipo, editarlos o eliminar dicho equipo de la Base de Datos. En la figura 63 se puede ver una imagen de esta interfaz.

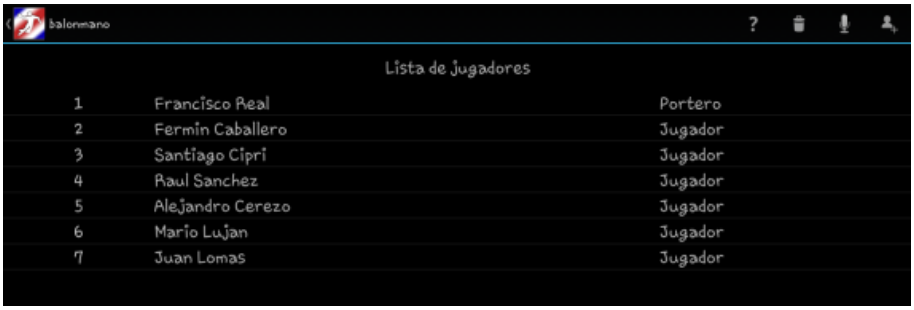



Figura 63. Interfaz Jugadores de Equipo

## 3.7.2 Arquitectura

- **Estructura UML**

Esta interfaz de ***Jugadores*** está compuesta por el archivo Jugadores.java, el cual tiene la siguiente estructura:

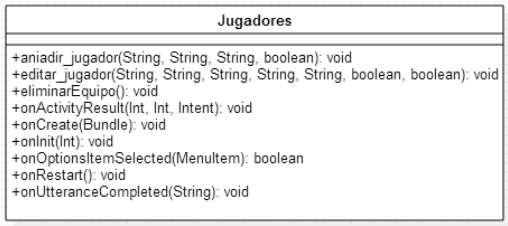



Figura 64. Diagrama UML de la clase Jugadores.java

- **Estructura de la ActionBar**

En este caso la interfaz de Equipos estará compuesta por una ActionBar que contiene por 4 iconos cuya funcionalidad se explica en la Figura 65.

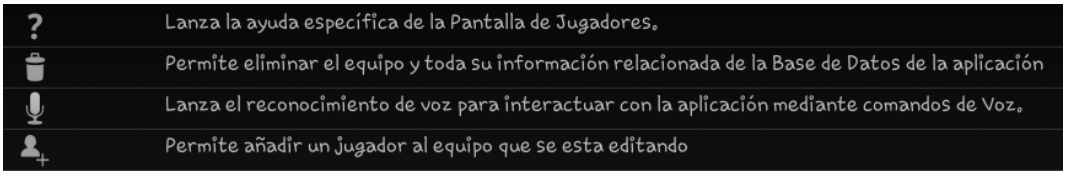



Figura 65. Botones de la ActionBar de la interfaz Jugadores

- **aniadir_jugador**

Es necesario recoger los valores de la interfaz *Datos de jugador*, desde la cual se introducen los datos de los jugadores. Si se cumplen los siguientes

requisitos se introducirá el nuevo valor del jugador en la Base de Datos asociada al equipo que corresponde:

- No se repite el nombre del Jugador para este mismo equipo en la Base de Datos

- No se repite el dorsal para ningún otro jugador del equipo en la Base de Datos.

Antes de la inserción en la Base de Datos se comprueba el valor *Posición* del jugador a añadir en la Base de Datos. Puesto que dependiendo de si la posición es *Jugador* o *Portero*, se incluirá la información en una tabla diferente de la BBDD. El diagrama asociado a la función *aniadir_jugador* es el siguiente:

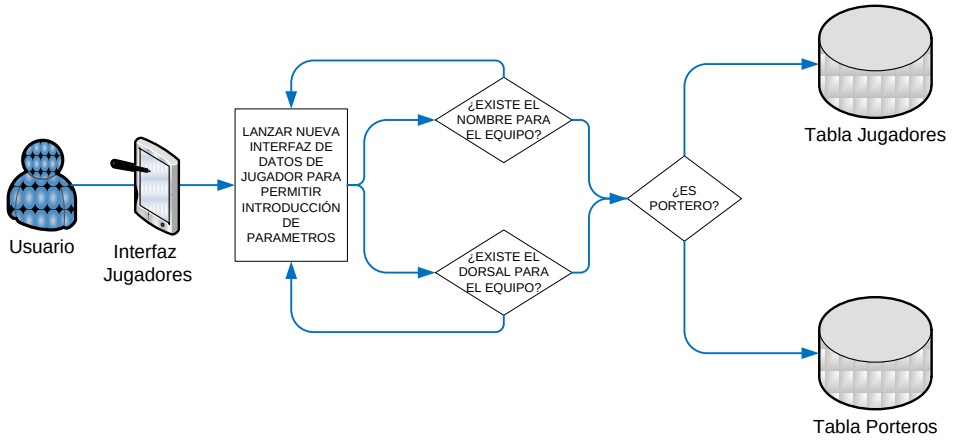



Figura 66. Diagrama aniadir_jugador

- **editar_jugador**

Para editar un jugador es necesario en primer lugar recoger los valores antiguos de la Base de Datos para mostrárselos al usuario. Posteriormente se le dará la opción de modificarlos a través de la interfaz *Datos del jugador*, y se recogerá la nueva información, de forma que es necesario sustituir los datos antiguos por los nuevos (Figura 67).

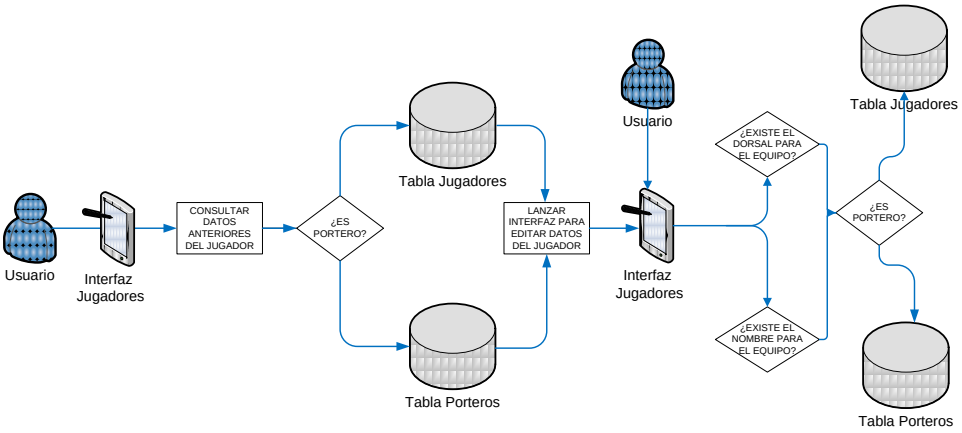



Figura 67. Diagrama editar_jugador

- **eliminarEquipo**

Para eliminar los datos del equipo es necesario no solo eliminar los datos del equipo de la Base de Datos, sino que además es necesario borrar todos los jugadores asociados a este equipo en la Base de Datos. Las relaciones existentes entre jugadores, porteros y equipos se pueden ver en la Figura 68.

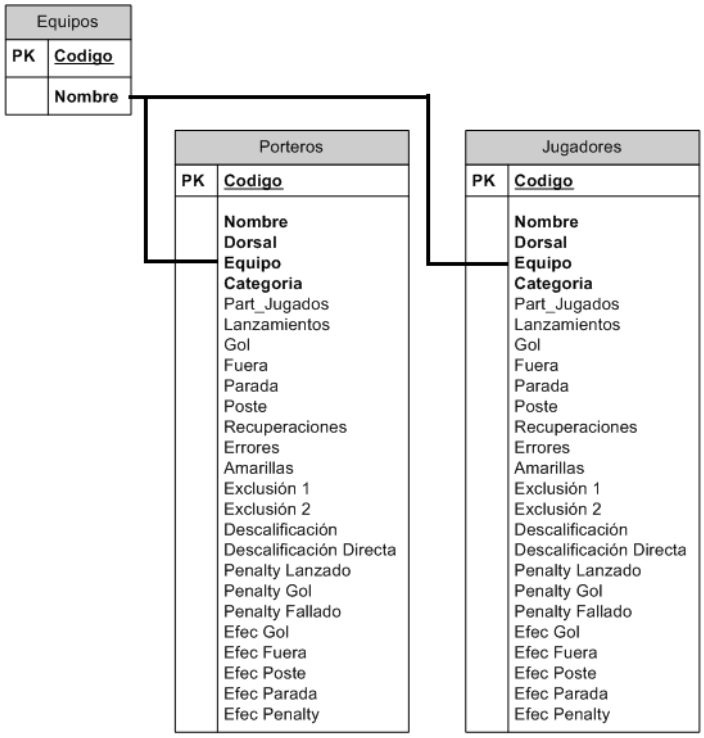



Figura 68. Relaciones BBDD Equipos con BBDD de Jugadores y Porteros

## 3.7.3 Comandos de Reconocimiento de Voz

Los valores que son reconocidos mediante el Reconocimiento de Voz de Android para esta interfaz son los siguientes:

| Comando de Voz | Descripción |
|---|---|
| Insertar/Añadir/Nuevo | Permite asociar un nuevo jugador a este equipo en la Base de Datos |
| Editar Jugador | Permite editar un jugador existente en la Base de Datos asociado al equipo |
| Eliminar | Permite eliminar un equipo y todos sus datos asociados en la Base de Datos |

Tabla 6. Comandos de voz para la interfaz *Jugadores*

## 3.7.4 Diagrama de flujo de la interfaz

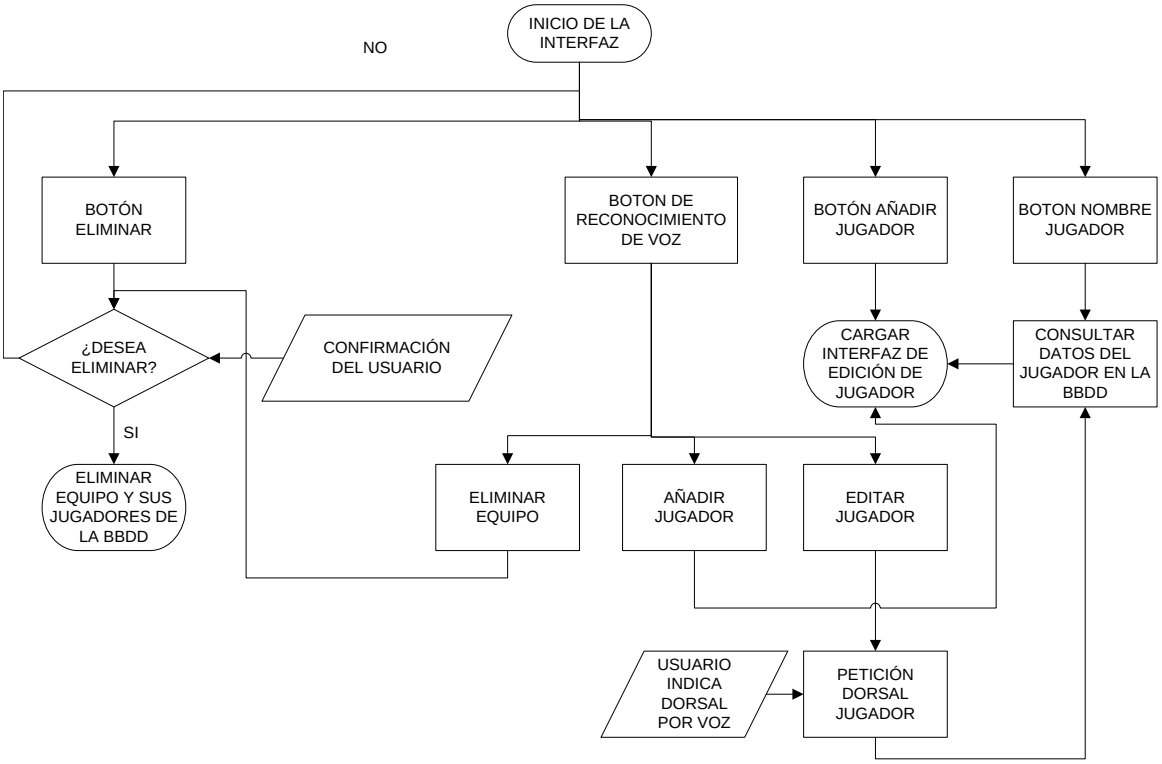



Figura 69. Diagrama de flujo de la Interfaz Jugadores

## 3.7.5 Escenarios de uso

En las Figuras 70, 71 y 72 se pueden ver ejemplos de diálogo entre el sistema y el usuario para la interfaz ***Jugadores***. Estos ejemplos representas los casos de *insertar un nuevo jugador, editar un jugador existente, eliminar un equipo.*

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Insertar |
| **S:** | **A continuación deberá introducir los datos del Nuevo Jugador** |

Figura 70. Ejemplo de escenario de uso en la interfaz Inserción

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Editar Jugador |
| **S:** | **Indique el dorsal del jugador que desea modificar** |
| U: | 1 |
| **S:** | **Se cargaran los datos del jugador de la Base de Datos** |

Figura 71. Ejemplo de escenario de uso en la interfaz Inserción

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Eliminar |
| **S:** | **¿Está seguro de que desea eliminar el equipo y todos sus componentes?** |
| U: | Si |
| **S:** | **Se ha eliminado correctamente el equipo de la Base de Datos** |

Figura 72. Ejemplo de escenario de uso en la interfaz Inserción

# 3.8 Sub-interfaz Datos de Jugador

## 3.8.1 Funcionalidad

A través de esta interfaz se pueden editar los datos de un jugador, ya sea de un jugador nuevo que se va a introducir en la Base de Datos o de un jugador ya existente del que se quieren modificar sus parámetros actuales. En la Figura 73 se puede ver una imagen de esta interfaz.

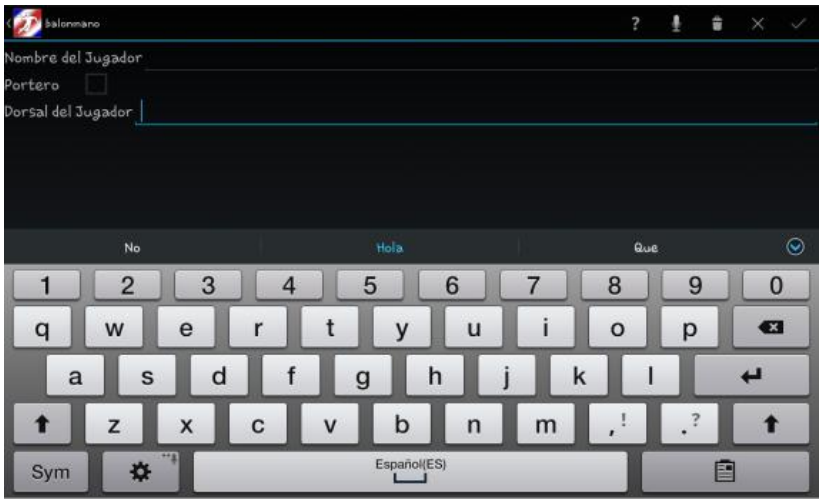



Figura 73. Interfaz Datos de Jugador

## 3.8.2 Arquitectura

- **Estructura XML**

Esta interfaz de inserción de equipo está compuesta por el archivo EditarJugador.java, el cual tiene la siguiente estructura:

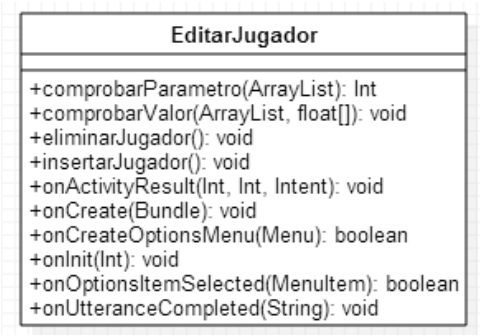



Figura 74. Diagrama UML de la clase EditarJugador.java

- **Estructura de la ActionBar**

En este caso la interfaz de ***Datos de Jugador*** estará compuesta por una ActionBar con 5 iconos cuya funcionalidad se explica en la Figura 75.

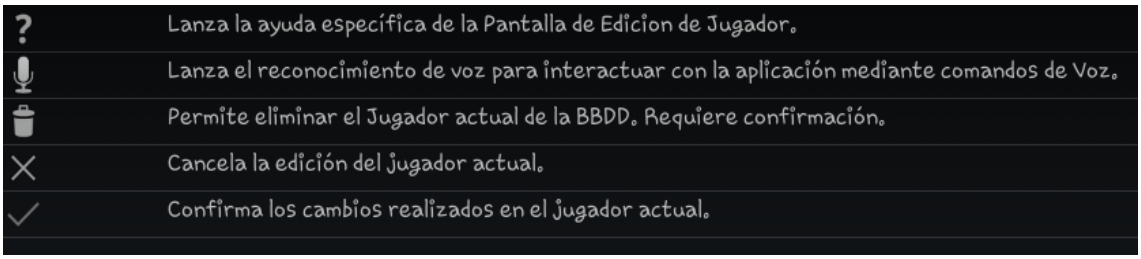



Figura 75. Botones de la ActionBar del Menú Datos de Jugador

- **comprobarParametro**

Esta función permite comprobar, si se está realizando la inserción a través de comandos de voz, que se han rellenado todos los parámetros necesarios o si falta alguno por rellenar.

- **comprobarValor**

Antes de devolver los valores insertados por el usuario a la interfaz de *Edición de Equipos*, que es la que realmente realiza las inserciones en la Base de Datos de la aplicación es necesario realizar una serie de comprobaciones para asegurarnos que los datos son correctos. Estas comprobaciones son:

  - Que se han introducido todos los datos necesarios del jugador. En este caso, el nombre y el Dorsal del jugador.

  - Que el dorsal contiene únicamente números, y no existen letras en este campo de texto que puedan crear errores al ser insertado en la Base de Datos.

- **eliminarJugador**

Al ser una operación crítica, que una vez realizada no tiene vuelta atrás, pues supone la eliminación definitiva de un jugador de la Base de Datos, es una acción que requiere confirmación por parte del usuario.

A través de esta función me aseguro de realizar este paso intermedio requiriendo la confirmación del usuario para realizar dicha acción.

- **insertarJugador**

Hace las comprobaciones oportunas a través de *comprobarValor* y si todo esta correcto manda los parámetros a la interfaz de *Edición de Equipos*, para que el jugador sea insertado en la Base de Datos, ya sea como un jugador nuevo, si es el caso, o para que se editen los valores de un jugador ya existente.

### 3.8.3 Comandos de Reconocimiento de Voz

Esta interfaz tiene un único comando de voz como tal, que se muestra en la Tabla 7. Esto es debido a que la interacción entre el sistema y el usuario para realizar la entrada de los datos del jugador se realiza a través de un diálogo marcado, en el que el usuario únicamente tendrá que dar los valores de los parámetros por los que se le pregunte y/o confirmar los datos de entrada.

| Comando de Voz | Descripción |
|---|---|
| Eliminar | Permite eliminar un equipo y todos sus datos asociados en la Base de Datos |
| Editar | Permite editar los parámetros del jugador actual en la BBDD |

Tabla 7. Comandos de voz para la interfaz *Datos de Jugador*

## 3.8.4 Diagrama de flujo de la interfaz

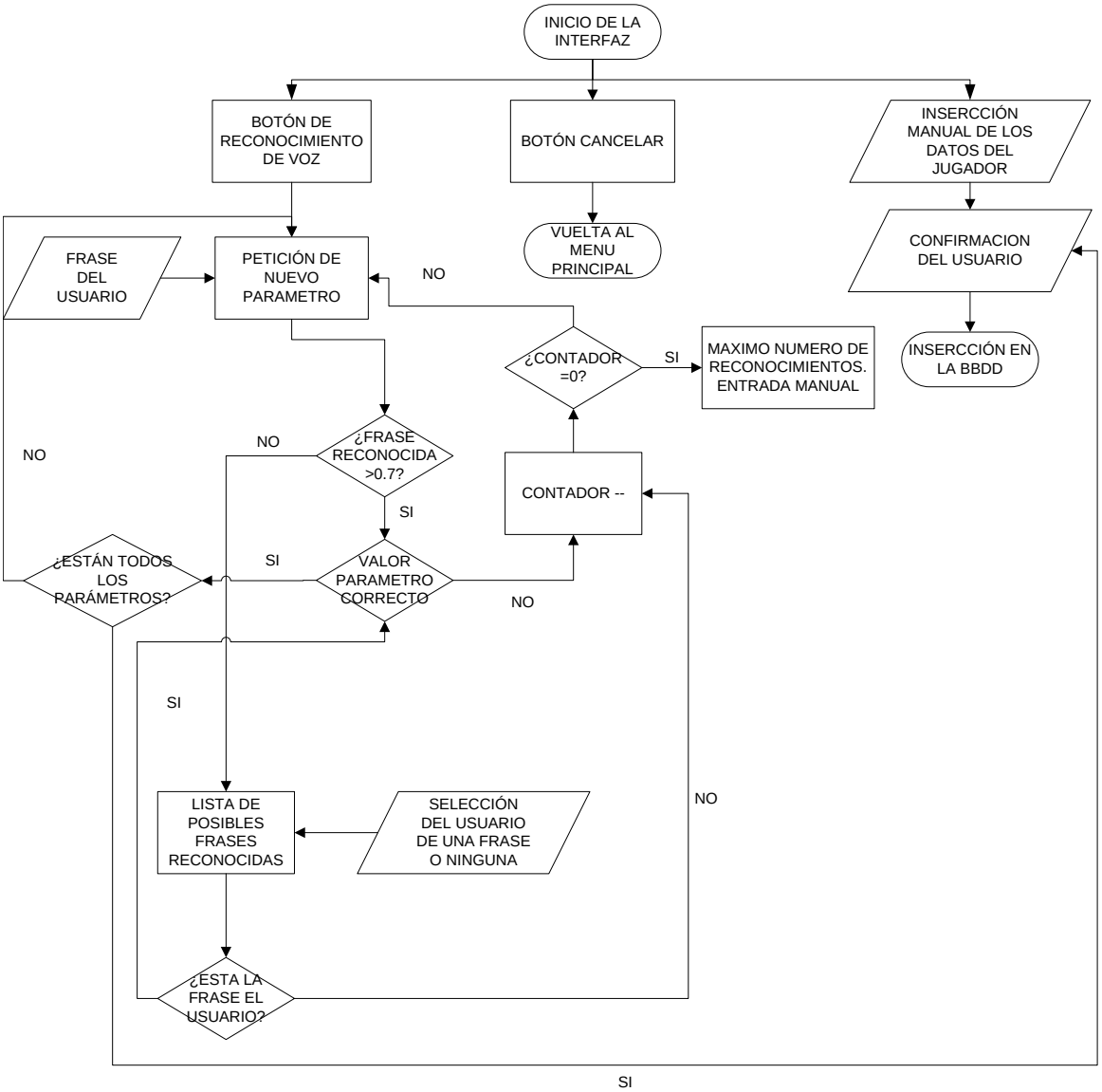



Figura 76. Diagrama de flujo de la interfaz *Datos de Jugador*

## 3.8.5 Escenarios de uso

| | |
|---|---|
| **S:** | **Indique el nombre del Jugador** |
| U: | Jonathan Caballero |
| **S:** | **¿Es correcto el valor Jonathan Caballero?** |
| U: | Si |
| **S:** | **¿La posición es portero?** |
| U: | Si |
| **S:** | **Indique el dorsal del jugador** |
| **U:** | 25 |
| **S:** | **¿Es correcto el valor 25?** |
| **U:** | Si |

Figura 77. Ejemplo de escenario de uso en la Interfaz de Edición de Jugador

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Eliminar |
| **S:** | **Indique el dorsal del jugador a eliminar** |
| U: | 25 |
| **S:** | **¿Está seguro que desea eliminar el jugador con dorsal 25?** |
| U: | Si |
| **S:** | **Se ha eliminado el jugador de la Base de Datos correctamente.** |

Figura 78. Ejemplo de escenario de uso en la Interfaz de Edición de Jugador

| | |
|---|---|
| **S:** | **¿Qué acción desea realizar?** |
| U: | Editar |
| **S:** | **¿Qué datos desea modificar del jugador?** |
| U: | Nombre |
| **S:** | **Indique el nombre del Jugador** |
| U: | Jonathan Caballero |
| **S:** | **¿Es correcto el valor Jonathan Caballero?** |
| U: | Si |
| **S:** | **¿Desea modificar otro parámetro?** |
| U: | No |

Figura 79. Ejemplo de escenario de uso en la Interfaz de Edición de Jugador

# 3.9 Interfaz de Estadísticas

## 3.9.1 Funcionalidad

A través de la interfaz *Estadísticas* se pueden consultar las estadísticas acumuladas de los jugadores y porteros de un equipo durante todos los partidos de una temporada. Para ello es posible elegir un equipo de entre los que se encuentran guardados en la BBDD, para ver los datos concretos de los jugadores pertenecientes a dicho equipo.

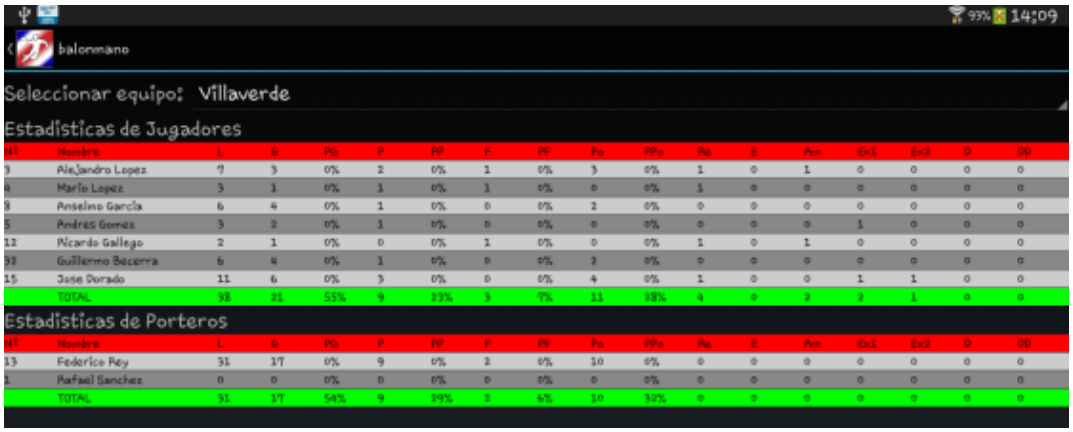



Figura 80. Subinterfaz de Estadísticas

## 3.9.2 Arquitectura

- **Estructura UML**

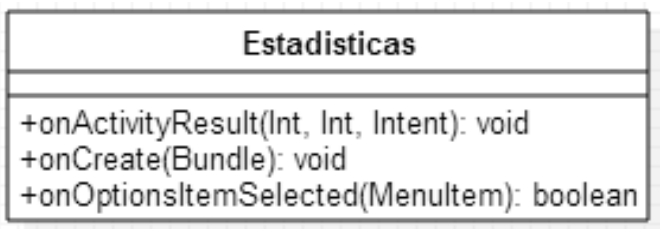



Figura 81. UML de la subinterfaz Estadísticas

- **Estructura de la actionbar**

Esta interfaz no contiene actionbar, puesto que no tiene comandos de voz asociados ni ninguna otra acción que requiera de la misma.

## 3.9.3 Comandos de Reconocimiento de Voz

Esta interfaz no contiene comandos de Reconocimiento de Voz, puesto que no existen acciones posibles sobre la misma. Se trata únicamente de una interfaz de visualización de información.

## 3.9.4 Escenarios de uso

Esta interfaz es únicamente de visualización de información y el tipo de interacción a realizar entre usuario y sistema mediante voz es la descripción de las estadísticas por parte del sistema de un jugador seleccionado por el usuario.

| | |
|---|---|
| **S:** | **El jugador Jonathan Caballero ha hecho 6 lanzamientos, 3 han sido gol, 1 parado, 1 fuera, 1 poste.** |

Figura 82. Escenarios de uso de la subinterfaz Estadísticas

# 3.10 Interfaz de Resultados

## 3.10.1 Funcionalidad

A través de la interfaz *Resultados* se pueden consultar los resultados de un equipo concreto pertenecientes a las estadísticas realizadas con el programa. Para ello es posible elegir un equipo de entre los que se encuentran guardados en la BBDD, para ver los datos concretos de dicho equipo.


Figura 83. Subinterfaz Resultados

## 3.10.2 Arquitectura

- **Estructura UML**

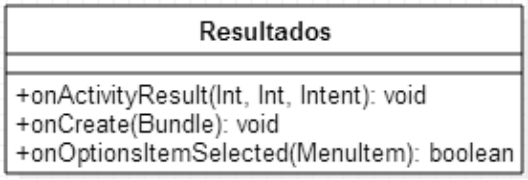



Figura 84. UML de la subinterfaz Resultados

- **Estructura de la actionbar**

Esta interfaz no contiene actionbar, puesto que no tiene comandos de voz asociados ni ninguna otra acción que requiera de la misma.

## 3.10.3 Comandos de Reconocimiento de Voz

Esta interfaz no contiene comandos de Reconocimiento de Voz, puesto que no existen acciones posibles sobre la misma. Se trata únicamente de una interfaz de visualización de información.

## 3.10.4 Escenarios de uso

Esta interfaz es únicamente de visualización de información y el tipo de interacción a realizar entre usuario y sistema mediante voz es la descripción de las estadísticas por parte del sistema de un jugador seleccionado por el usuario.

| | |
|---|---|
| **S:** | **El jugador Jonathan Caballero ha hecho 6 lanzamientos, 3 han sido gol, 1 parado, 1 fuera, 1 poste.** |

Figura 85. Escenario de Uso de la subinterfaz Resultados

# 3.11 Interfaz de Nuevo Partido

La interfaz de Nuevo Partido será la interfaz principal de la aplicación. Es aquella que una vez introducida la información de Datos y Jugadores en la Base de Datos permitirá al usuario gestionar las estadísticas en tiempo real de un partido de balonmano, que al fin y al cabo es la función principal del diseño de esta aplicación.

En la Figura 86 se pude observar las diferentes subinterfaces que tiene la línea de la Interfaz *Nuevo Partido* y cada una de las ramas por las que puede navegar el usuario partiendo del *Menú Principal.*


Figura 86. Subinterfaces y árbol de Navegación de la Interfaz *Nuevo Partido*

# 3.12 Sub-interfaz Elección de equipos

## 3.12.1 Funcionalidad

En esta interfaz se dará comienzo al primer paso para iniciar un nuevo partido entre dos equipos. Para poder llegar hasta ella necesitamos que al menos existan 2 equipos en la Base de Datos, que es el número mínimo lógico de equipos necesario para comenzar un partido.

Se permitirá elegir en este caso, tanto mediante intervención táctil del usuario como por comandos de voz los dos equipos que participarán en el partido, además de cuál de ellos actúa como visitante y cuál como local.

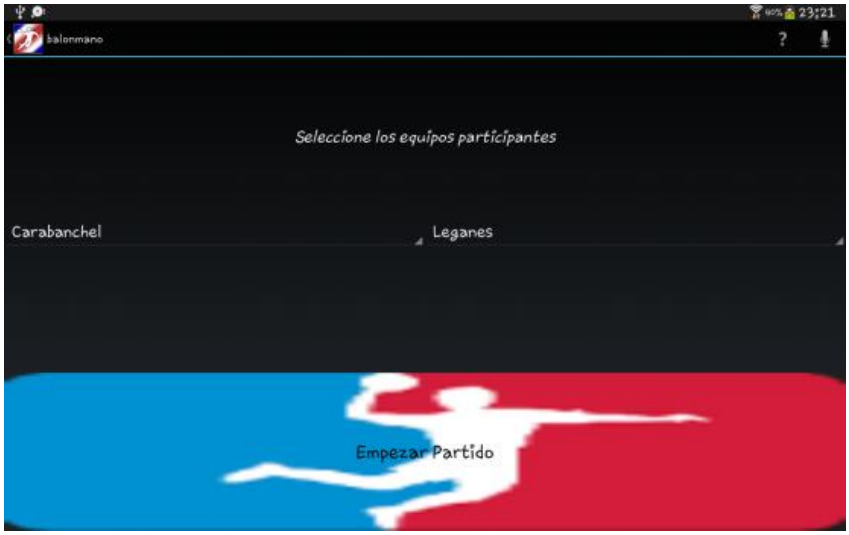



Figura 87. Interfaz de elección de equipos.

## 3.12.2 Arquitectura

- **Estructura XML**

Esta interfaz de *Selección de Equipos* está compuesta por el archivo NuevoPartido.java, el cual tiene la siguiente estructura:

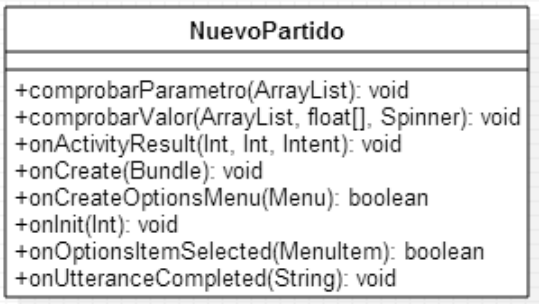



Figura 88. UML de la subinterfaz de Elección de Equipos

- **Estructura de la ActionBar**

En este caso la interfaz de *Selección de Equipos* estará compuesta por una ActionBar que contiene 2 iconos cuya funcionalidad se explica en la Figura 89.

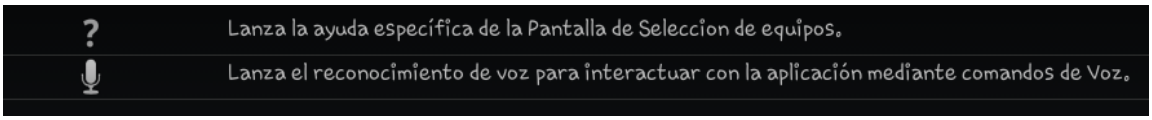



Figura 89. Botones de la ActionBar del Menú Nuevo Partido

- **comprobarValor**

Antes de comenzar el partido es necesario realizar una pequeña comprobación en esta pantalla para asegurar que el inicio del partido es correcto. Esta comprobación se trata únicamente de asegurar que el equipo local y el equipo visitante no coinciden en nombre.

### 3.12.3 Comandos de Reconocimiento de Voz

| Comando de Voz | Descripción |
|---|---|
| Local/Visitante | Permite indicar el equipo |

Tabla 8. Comandos de voz para la interfaz *Selección de Equipo*

## 3.12.4 Diagrama de Flujo de la Interfaz

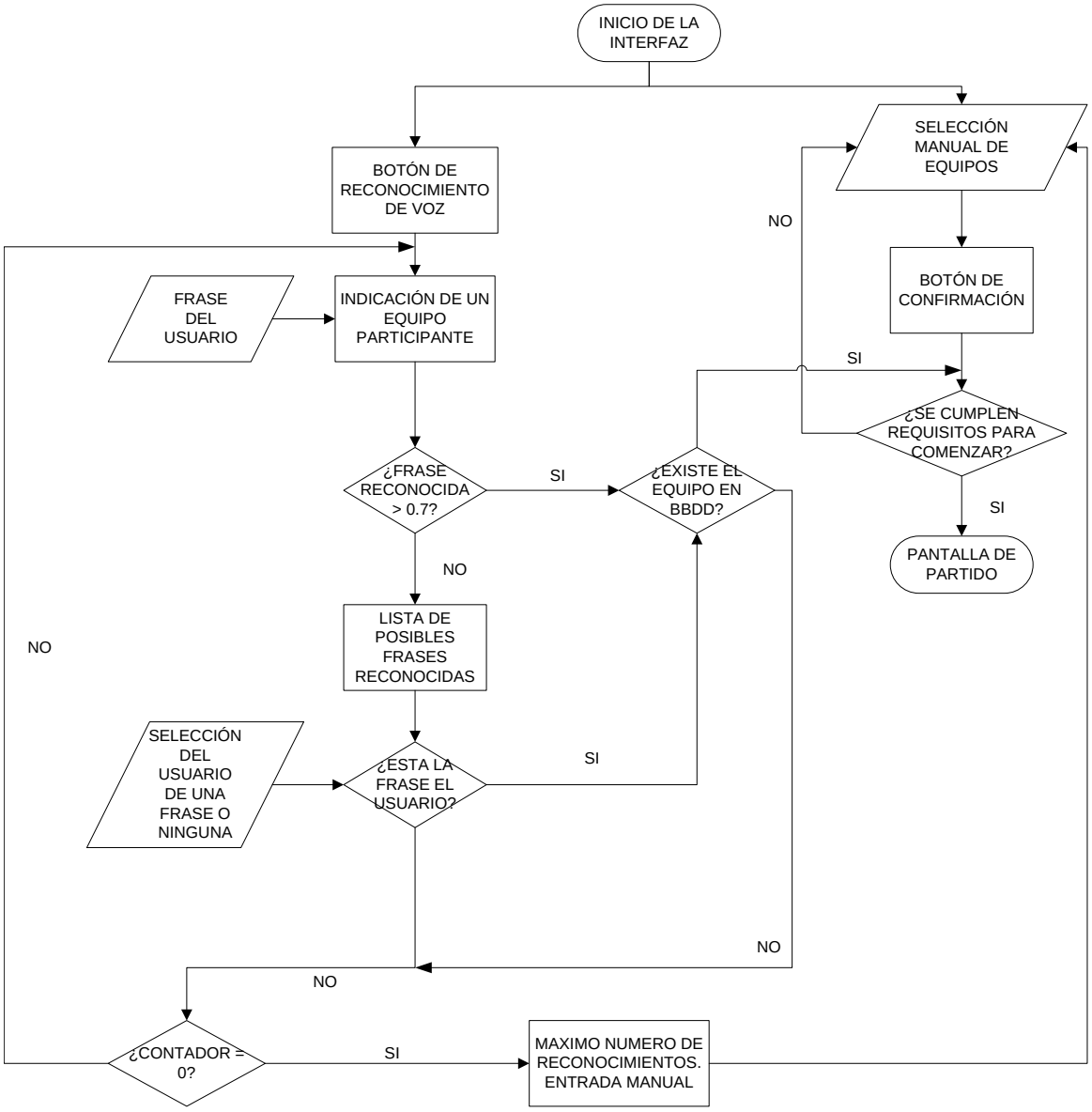



Figura 90, Diagrama de flujo de la Interfaz Selección de equipos

## 3.12.5 Escenarios de uso

| | |
|---|---|
| **S:** | **¿Qué equipo desea seleccionar?** |
| U: | Local |
| **S:** | **Diga el nombre del equipo local** |
| U: | Villaverde |
| **S:** | **¿Es correcto el nuevo valor Villaverde?** |
| U: | Si |
| **S:** | **Diga el nombre del equipo visitante** |
| U: | Leganés |
| **S:** | **¿Es correcto el nuevo valor Leganés?** |
| U: | Si |
| **S:** | **Comenzará un nuevo partido entre Villaverde y Leganés.** |

Figura 91. Ejemplo de escenario de uso en la Interfaz de Selección de Equipos

# 3.13 Sub-interfaz Elección de porteros

## 3.13.1 Funcionalidad

En esta interfaz se realizará la siguiente elección previa a comenzar el partido, que será la de elegir el portero titular de ambos equipos antes de comenzar el partido. Esta acción se hace necesaria debido a que de forma habitual un equipo tiene varios porteros y, como las estadísticas de los porteros se realizan de forma automática en función de los lanzamientos de los jugadores del equipo contrario seleccionados en la aplicación.

A esta pantalla (Figura 92) además se podrá volver en cualquier momento a través de una barra lateral o *DrawerLayout* que permitirá cambiar el portero del equipo en cualquier momento del partido ante cualquier sustitución que se pueda producir en cualquiera de los dos equipos.

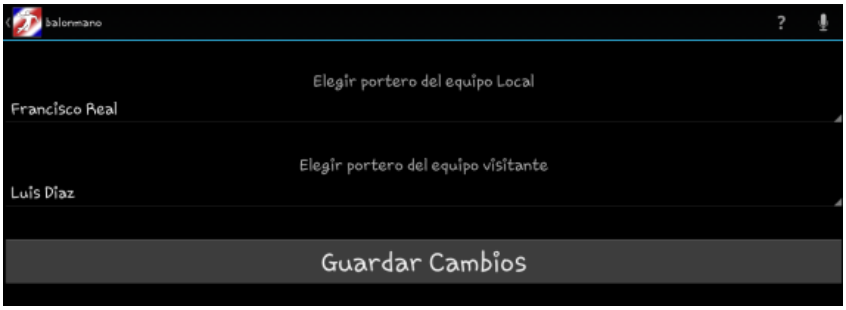



Figura 92. Subinterfaz porteros

## 3.13.2 Arquitectura

- **Estructura XML**

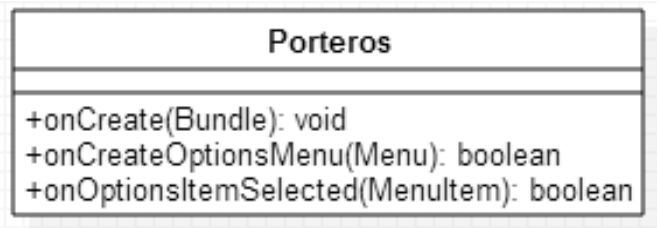



Figura 93. Diagrama UML de la clase Porteros.java

# 3.14 Sub-interfaz Pantalla de partido

## 3.14.1 Funcionalidad

Esta es la interfaz principal que utilizará el usuario para introducir las estadísticas en tiempo real de los partidos (Figura 94). A través de la misma podrá introducir tanto a través de comandos de voz como a través de

pulsaciones táctiles de la pantalla toda la información que vaya ocurriendo durante el partido, tal como lanzamientos, errores de los jugadores, amonestaciones, etc.


Figura 94. Subinterfaz Pantalla de Partido

Además a través de esta pantalla podemos acceder a información adicional de los datos del partido a través de una barra lateral, a través de lo que se conoce como un DrawerLayout. Esta información complementaria es: Lista ordenada de las acciones transcurridas durante el partido, estadísticas hasta el momento actual del equipo local y visitante, pantalla de selección de porteros de equipo. La visualización de la interfaz con esta barra lateral se puede ver en la Figura 95.

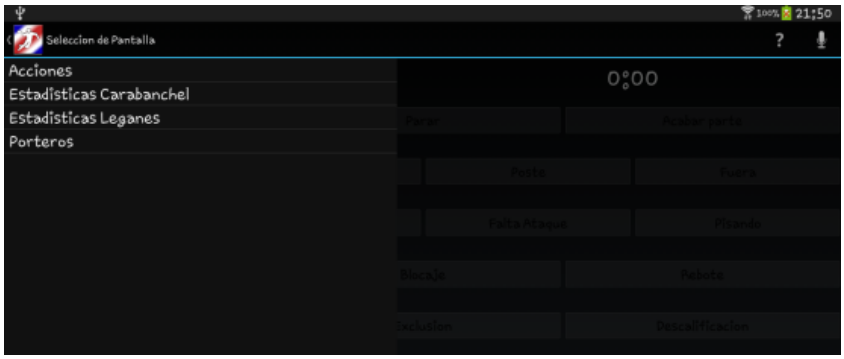



Figura 95. Barra Lateral de la Subinterfaz Pantalla de Partido

## 3.14.2 Arquitectura

- **Estructura UML**

En la Figura 96 podemos ver el Diagrama UML de la clase PantallaPartido.java

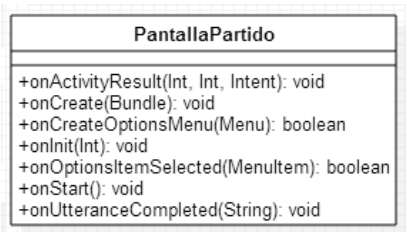



Figura 96. Diagrama uml de la clase PantallaPartido

- **Reconocimiento de voz**

Esta es una pantalla especial en cuanto al reconocimiento de voz se refiere. El motivo es debido a que no es posible realizar la confirmación de las acciones por parte del usuario. Puesto que si se espera confirmación por parte del usuario se podrían suceder nuevas acciones del partido y no daría tiempo a registrarlas.

- **Estructura de la ActionBar**

En este caso la interfaz de *Selección de Equipos* estará compuesta por una ActionBar que contiene 2 iconos cuya funcionalidad se explica en la Figura 97.

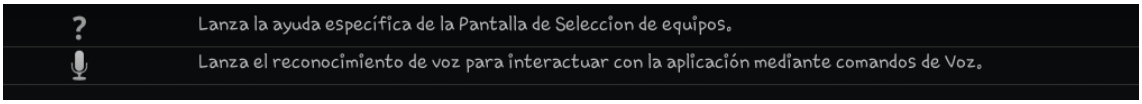



Figura 97. Botones de la ActionBar del Menú Nuevo Partido

## 3.14.3 Comandos de Reconocimiento de Voz

| Comando de Voz | Descripción |
|---|---|
| Comenzar | Permite iniciar un partido o reiniciarlo en caso de pausa |
| Parar | Permite parar un partido que está en curso |
| Gol | Para añadir una acción de gol a las estadísticas actuales |
| Parada | Para añadir una acción de parada a las estadísticas actuales |
| Poste | Para añadir una acción de poste a las estadísticas actuales |
| Fuera | Para añadir una acción de fuera a las estadísticas actuales |
| Pasos | Para añadir un error de jugador por pasos a las estadísticas actuales |
| Dobles | Para añadir un error de jugador por dobles a las estadísticas actuales |
| Ataque | Para añadir un error de jugador por falta en ataque a las estadísticas actuales |
| Pisando | Para añadir un error de jugador por pisando a las estadísticas actuales |
| Recuperación | Para añadir un acierto en una acción defensiva del jugador por recuperación a la estadísticas actuales |
| Blocaje | Para añadir un acierto en una acción defensiva del jugador por blocaje a la estadísticas actuales |
| Rebote | Para añadir un acierto en una acción defensiva del jugador por rebote a la estadísticas actuales |
| Amarilla | Para añadir una amonestación por amarilla a las estadísticas actuales |
| Exclusión | Para añadir una amonestación por exclusión a las estadísticas actuales |
| Descalificación | Para añadir una amonestación por descalificación a las estadísticas actuales |

Tabla 9. Comandos de voz para la interfaz *Selección de Equipo*

### 3.14.4 Diagrama de flujo de la interfaz

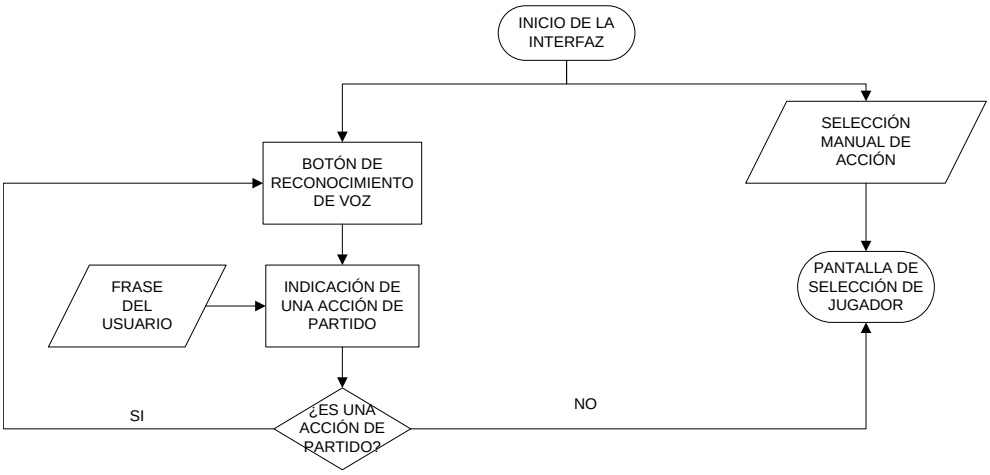



Figura 98. Diagrama de Flujo de la subinterfaz Pantalla de Partido

### 3.14.5 Escenarios de Uso

No existen escenarios de uso como tal para esta pantalla. Ya que no hay interacción sistema-usuario, si no que únicamente es el usuario el que debe indicar la acción a insertar en la BBDD al ser producida durante el partido.

## 3.15 Sub-interfaz Elección Jugador que realiza la acción

### 3.15.1 Funcionalidad

Esta interfaz le sirve al usuario para seleccionar el jugador que ha realizado una acción durante el partido y que ha sido seleccionada previamente. En ella se presentan todos los jugadores de campo de ambos equipos mostrándose su dorsal y el nombre del jugador para identificarlos fácilmente.

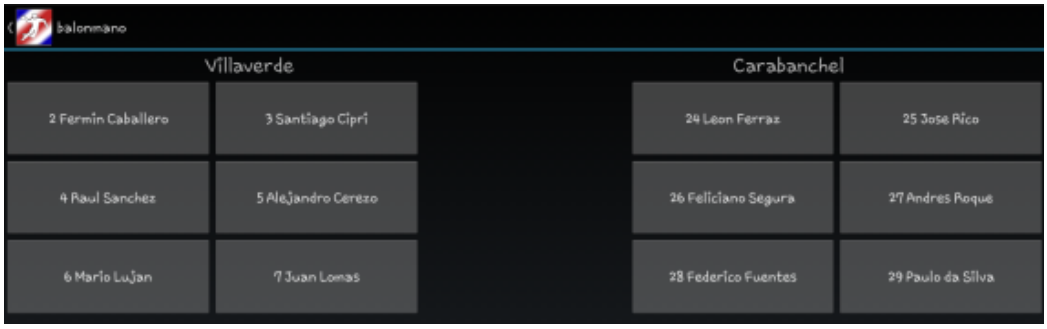



Figura 99. Subinterfaz Elección de Jugador

## 3.15.2 Arquitectura

- **Estructura UML**

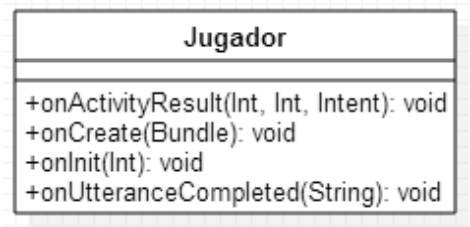



Figura 100. Estructura UML Elección de Jugador

- **Estructura de la actionbar**

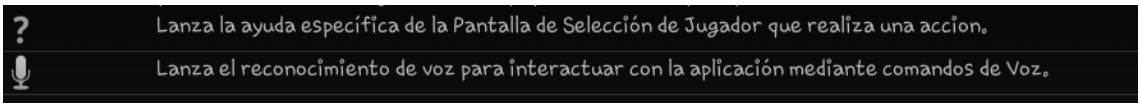



Figura 101. Action Bar Elección de Jugador

- **Muestra de jugadores**

Es necesario realizar una consulta para sacar de la base de Datos el nombre y dorsal de todos los jugadores de cada uno de los dos equipos participantes en el encuentro para poder mostrarlos de forma ordenada.

Para facilitarle al usuario encontrar el jugador a seleccionar, cuando se extraen los nombres de la base de datos se aplica un algoritmo de ordenación por dorsal, de forma que se muestran en pantalla ordenador de menos a mayor dorsal.

Es necesario sacar la información tanto de la tabla de jugadores de campo como de la tabla de porteros, ya que, aunque no es habitual, los porteros en contadas situaciones también realizan acciones asociadas a jugadores de campo, como por ejemplo puede ser un lanzamiento a portería.

Tras la selección del jugador, la acción propuesta debe ser registrada en la base de datos. Para guardar la información que conlleva una acción es necesario modificar varias tablas en la BBDD de la aplicación tales como:

- BBDD *Partido<LocalVisitante>* Tabla *Jugadores*: Se modificará para el jugador que ha realizado la acción la estadística correspondiente.

- BBDD *Partido<LocalVisitante>* Tabla *Porteros*: Se modificará para el portero que esta seleccionado como “portero actual en campo” y que juega en el equipo contrario al jugador para el que hemos marcado la nueva acción la acción la estadística correspondiente.

- BBDD *Partido<LocalVisitante>* Tabla *Partido:* Se guarda una descripción de la acción seleccionada para llevar un registro de todo lo que transcurre durante el partido.

## 3.15.3 Comandos de Reconocimiento de Voz

| Comando de Voz | Descripción |
|---|---|
| Local/visitante | Permite indicar de que equipo es el jugador que ha realizado la acción seleccionada |
| Dorsal/Nombre | Después de indicar el equipo del jugador, se puede identificar dicho jugado mediante su nombre o dorsal. |

Tabla 10. Comandos de voz para la interfaz *Selección de Equipo*

## 3.15.4 Diagrama de flujo de la interfaz

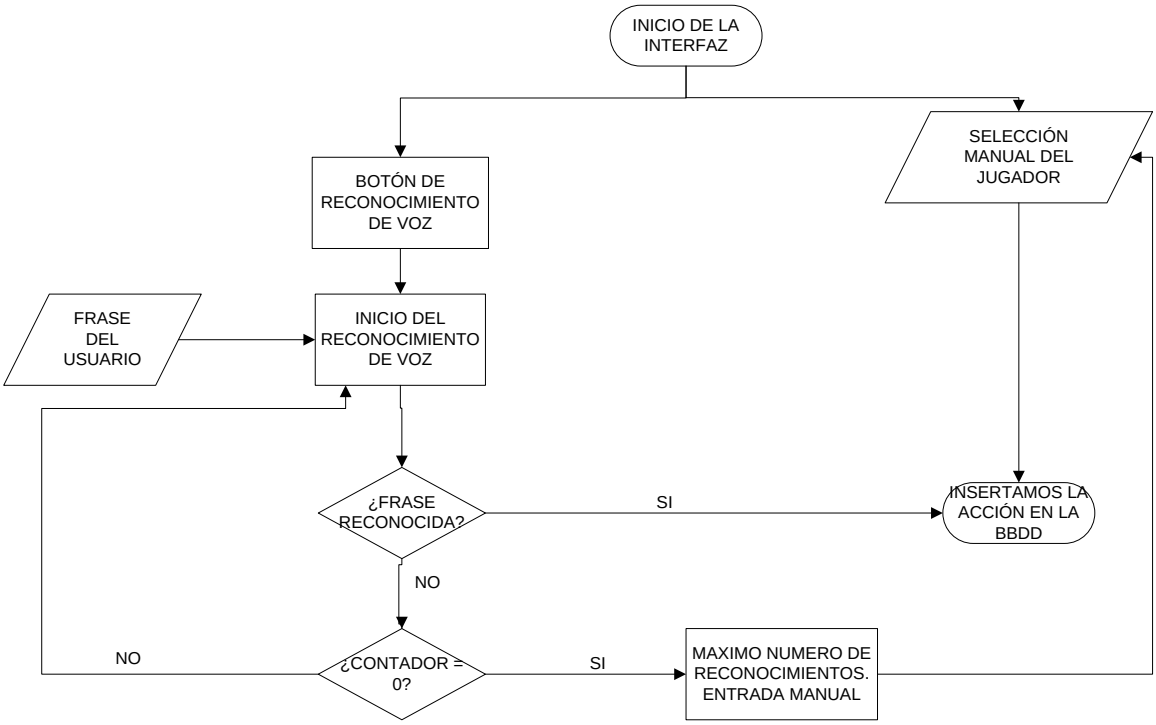



Figura 102. Diagrama de Flujo de la Interfaz Jugador

## 3.15.5 Escenarios de uso

| | |
|---|---|
| **S:** | **Seleccione el jugador que ha realizado la acción** |
| U: | Local Jonathan Caballero |

Figura 103. Escenario de Uso 1 Elección de Jugador

| | |
|---|---|
| **S:** | **Seleccione el jugador que ha realizado la acción** |
| U: | Visitante 10 |

Figura 104. Escenario de Uso 2 Elección de Jugador

# 3.16 Sub-interfaz Acciones del partido

## 3.16.1 Funcionalidad

Esta interfaz permite visualizar en tiempo real en un partido una lista de todas las acciones que han transcurrido hasta el momento durante el mismo.

Se podrán identificar en las acciones el minuto en el que se realizó la misma, si fue por parte del equipo local o visitante, el jugador que la llevó a cabo y si supone un cambio en el marcador del encuentro indicando el valor actual de dicho marcador.

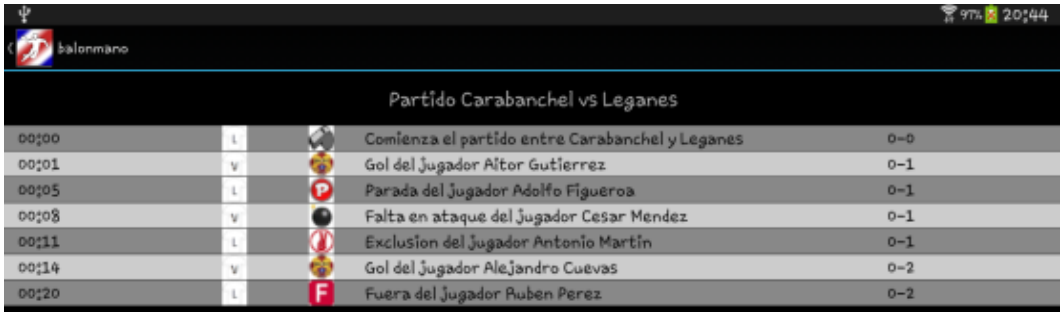



Figura 105. Subinterfaz de Acciones

## 3.16.2 Arquitectura

- **Estructura UML**

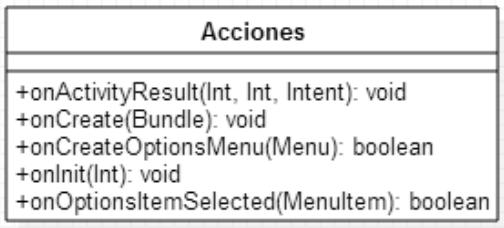



Figura 106. UML de la Subinterfaz de Acciones

- **Estructura de la actionbar**

Esta interfaz no contiene actionbar, puesto que no tiene comandos de voz asociados ni ninguna otra acción que requiera de la misma.

## 3.16.3 Comandos de Reconocimiento de Voz

Esta interfaz no contiene comandos de Reconocimiento de Voz, puesto que no existen acciones posibles sobre la misma. Se trata únicamente de una interfaz de visualización de información.

### 3.16.4 Escenarios de uso

Aunque esta interfaz no tiene comandos de voz asociados, si que existe una interacción mediante voz entre el usuario y el sistema. Ya que el usuario puede seleccionar una acción para que el sistema le dé información de la misma mediante voz.

| | |
|---|---|
| **S:** | **Tiempo 10 minutos 32 segundos, Gol del jugador Jonathan Caballero del equipo Villaverde, Resultado 6 a 3.** |

Figura 107. Escenario de Uso de la Subinterfaz de Acciones

## 3.17 Sub-interfaz Estadísticas Jugadores durante el partido

### 3.17.1 Funcionalidad

A través de la interfaz *Estadísticas de Jugadores* se puede consultar en tiempo real las estadísticas de los jugadores y porteros de un equipo en el momento actual del partido y que incluye todas las acciones realizadas por los mismos durante el transcurso de dicho partido.


| Nº | Nombre | L | G | PG | P | PP | F | PF | Po | PPo | Pe | E | Am | Ex1 | Ex2 | D | DD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Fermín Caballero | 0 | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | Santiago Cipri | 0 | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | Raul Sanchez | 0 | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 5 | Alejandro Cerezo | 1 | 1 | 100% | 0 | 0% | 0 | 0% | 0 | 0% | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | Mario Lujan | 1 | 1 | 100% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Juan Lomas | 0 | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 0% | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | TOTAL | 2 | 2 | 100% | 0 | 0% | 0 | 0% | 0 | 0% | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| Nº | Nombre | L | G | PG | P | PP | F | PF | Po | PPo | | | | | | | |
| 1 | Francisco Real | 6 | 3 | 50% | 2 | 33% | 0 | 0% | 2 | 33% | | | | | | | |
| | TOTAL | 6 | 3 | 50% | 2 | 33% | 0 | 0% | 2 | 33% | | | | | | | |

Figura 108. Subinterfaz de Estadísticas de Equipo

### 3.17.2 Arquitectura

- **Estructura UML**

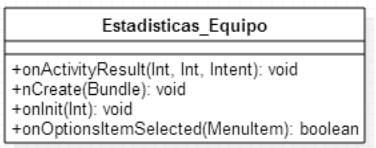



Figura 109. UML de Estadísticas de Equipo

- **Estructura de la actionbar**

Esta interfaz no contiene actionbar, puesto que no tiene comandos de voz asociados ni ninguna otra acción que requiera de la misma.

### 3.17.3 Comandos de Reconocimiento de Voz

Esta interfaz no contiene comandos de Reconocimiento de Voz, puesto que no existen acciones posibles sobre la misma. Se trata únicamente de una interfaz de visualización de información.

### 3.17.4 Escenarios de uso

Al igual que la interfaz de acciones, esta interfaz es únicamente de visualización de información y el tiempo de interacción a realizar entre usuario y sistema mediante voz es la descripción de las estadísticas por parte del sistema de un jugador seleccionado por el usuario.

| | |
|---|---|
| **S:** | **El jugador Jonathan Caballero ha hecho 6 lanzamientos, 3 han sido gol, 1 parado, 1 fuera, 1 poste.** |

Figura 110. Escenario de Uso de la Subinterfaz de Estadísticas de Jugador

# Capítulo 4. Pruebas del sistema

En este capítulo voy a especificar las pruebas realizadas sobre el sistema. En primer lugar indicaré la forma de proceder a la hora de realizar un plan de pruebas que resulte efectivo, posteriormente indicaré las diferentes fases en que se ha separado dicho plan de pruebas y finalmente expondré los casos de prueba.

## 4.1 Introducción

Se realizan diferentes tipos de pruebas para realizar una validación más completa del sistema. Las pruebas estarán dividas por tanto en fases de la siguiente forma:

- Pruebas de flujo de la aplicación: Me permitirán probar el correcto flujo entre las distintas interfaces de la aplicación. De forma que pueda asegurar la correcta transición entre ellas y ante las diferentes formas de navegación posible, es decir ante la posibilidad de cambiar de navegación a través de comandos de voz o mediante comandos táctiles. Así como especificar los diferentes errores posibles ante el cambio entre interfaces y el tratamiento que se hace de los mismos.

- Pruebas de funcionalidad de la aplicación: A través de estas pruebas verificaré el funcionamiento de cada una de las interfaces de la aplicación así como los objetivos por el que fue creada cada una de ellas. Se presentará también los posibles errores localizados que puedan producirse al tratar cada una de las funcionalidades a probar así como la solución planteada para su tratamiento.

## 4.2 Plantilla de pruebas del sistema

En este apartado realizaré una breve descripción de la plantilla que utilizaré para registrar todas las pruebas a realizar en el sistema. Para el análisis identificaré cada una de las pruebas en una tabla que contendrá los siguientes campos:

- **Identificador:** Identifica de forma unívoca cada una de las pruebas del sistema. El formato para la nomenclatura de las pruebas será P**XXX-YY** donde **XXX** identificará la tipología de las pruebas entre:

- **Flu:** Pruebas de flujo de las interfaces de la aplicación.
- **Fun:** Pruebas funcionales de la aplicación.

**YY** identificará el número de la prueba dentro de las de su grupo.

- **Descripción de línea principal:** Descripción del objetivo principal de la prueba.
- **Procedimiento de la línea principal:** Procedimiento a realizar para llevar a cabo la línea principal de la prueba.
- **Resultado de la línea principal:** Resultado obtenido al ejecutar la línea principal de la prueba.
- **Descripción de la variante n:** Descripción del objetivo de una prueba alternativa relacionada con la realizada en la línea principal.
- **Procedimiento de la variante n:** Procedimiento a realizar para llevar a cabo la variante de la prueba descrita.
- **Resultado de la variante n:** Resultado obtenido al ejecutar la prueba alternativa.
- **Gestión de errores posibles:** Indicación de posibles errores que podría encontrar el usuario al realizar las acciones descritas en las pruebas.
- **Errores controlados:** Control que se realiza por parte de la aplicación de los errores posibles que se puede encontrar el usuario.

En la Tabla 11 se muestra un ejemplo de la plantilla que se ha seguido para la recogida de la información del resultado de la realización de las pruebas de la aplicación.

| Identificador | PXXX-YY |
|---|---|
| Descripción línea principal | |
| Procedimiento de la línea principal | |
| Resultado de la línea principal | |
| Variante 1 de la prueba | |
| Procedimiento de la variante 1 | |
| Resultado de la variante 1 | |
| ….. | |
| Variante n de la prueba | |
| Procedimiento de la variante 2 | |
| Resultado de la variante n | |
| Gestión de errores posibles | |
| ¿Errores controlados? | |

Tabla 11. Plantilla de Pruebas

# 4.3 Pruebas realizadas

## 4.3.1 Pruebas de flujo

Estas pruebas se definirán con el identificador 'PFlu-YY'. Su principal objetivo será probar el funcionamiento de las interfaces de la aplicación y su correcto flujo.

| Identificador | PFlu-01 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Edición de Equipos*. Modo táctil. |
| **Procedimiento de la línea principal** | Iniciar la aplicación.<br>Pulsar el botón de *Edición de Equipos* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Edición de Equipos*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Iniciar la aplicación.<br>Utilizar el comando "Editar" para realizar la transición. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Edición de Equipos* a la interfaz *Menú Principal*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Edición de Equipos.*<br>Utilizar el botón *home* de la ActionBar o el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil. |

Tabla 4. PFlu-01

| Identificador | PFlu-02 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Selección de Equipos*. Modo táctil. |
| **Procedimiento de la línea principal** | Iniciar la aplicación.<br>Pulsar el botón de *Nuevo Partido* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Nuevo Partido*. Comandos de |

| | |
|---|---|
| | Voz. |
| **Procedimiento de la variante 1** | Iniciar la aplicación.<br>Utilizar el comando "Nuevo" para realizar la transición. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Selección de Equipos* a la interfaz *Menú Principal*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Nuevo Partido.*<br>Utilizar el botón *home* de la ActionBar o el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz<br>Error 4: Número suficiente de equipos en la BBDD |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil.<br>RespErr4: En caso de no haber al menos 2 equipos en la BBDD de la aplicación el sistema indica un mensaje de voz para solventar esta situación. |

Tabla 5. PFlu-02

| Identificador | PFlu-03 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Resultados*. Modo táctil. |
| **Procedimiento de la línea principal** | Iniciar la aplicación.<br>Pulsar el botón de *Resultados* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Resultados*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Iniciar la aplicación.<br>Utilizar el comando "Resultados" para realizar la transición. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Resultados* a la interfaz *Menú Principal*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Resultados*<br>Utilizar el botón *home* de la ActionBar o el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz |

| | |
|---|---|
| | Error 4: Número suficiente de equipos en la BBDD |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil.<br>RespErr4: En caso de no haber al menos 2 equipos en la BBDD de la aplicación el sistema indica un mensaje de voz para solventar esta situación. |

Tabla 6. PFlu-03

| Identificador | PFlu-04 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Estadísticas*. Modo táctil. |
| **Procedimiento de la línea principal** | Iniciar la aplicación.<br>Pulsar el botón de *Estadísticas* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Menú Principal* a la interfaz de *Estadísticas*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Iniciar la aplicación.<br>Utilizar el comando "Estadísticas" para realizar la transición. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Estadísticas* a la interfaz *Menú Principal*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Estadísticas.*<br>Utilizar el botón *home* de la ActionBar o el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz<br>Error 4: Número suficiente de equipos en la BBDD |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil.<br>RespErr4: En caso de no haber al menos 2 equipos en la BBDD de la aplicación el sistema indica un mensaje de voz para solventar esta situación. |

Tabla 7. PFlu-04

| Identificador | PFlu-05 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Edición de Equipos* a la interfaz de *Inserción*. Modo táctil. |
| **Procedimiento de la línea principal** | Desde la interfaz de *Edición de Equipos*.<br>Pulsar el icono de la ActionBar de *Añadir Equipo* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Edición de Equipos* a la interfaz de *Inserción*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde la interfaz de *Edición de Equipos*.<br>Utilizar cualquiera de los comandos "Insertar/Añadir/Nuevo" para realizar la transición. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Inserción* a la interfaz *Edición de Equipos*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Inserción.*<br>Utilizar el botón *home* de la ActionBar;<br>o el botón *atrás* de Android;<br>o el botón cancelar de la Action Bar;<br>o el botón confirmar de la Action Bar dejando el campo nombre vacío. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil. |

Tabla 8. PFlu-05

| Identificador | PFlu-06 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Edición de Equipos* a la interfaz de *Jugadores*. Modo táctil. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Edición de equipos*.<br>Pulsar el nombre del equipo del que se desea modificar los jugadores dentro de la lista de equipos. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Edición de Equipos* a la interfaz de *Jugadores*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde la interfaz de *Edición de Equipos*.<br>Utilizar el comando "Editar".<br>Indicar el número de posición que ocupa el equipo del que se desea modificar los jugadores dentro de la lista de equipos. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de |

| | |
|---|---|
| | *Jugadores* a la interfaz *Edición de Equipos*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Inserción.*<br>Utilizar el botón *home* de la ActionBar o el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil. |

Tabla 9. PFlu-06

| Identificador | PFlu-07 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Jugadores* a la interfaz de *Edición de Jugador*. Modo táctil. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Jugadores*.<br>Pulsar el nombre del jugador del que se desea modificar los datos. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Jugadores* a la interfaz de *Edición de Jugador*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde la interfaz de *Jugadores*.<br>Utilizar el comando "Editar".<br>Indicar el dorsal del jugador o el nombre del que se desea modificar los datos. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Edición de Jugador* a la interfaz *Jugadores*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Inserción.*<br>Utilizar el botón *home* de la ActionBar;<br>o el botón *atrás* de Android;<br>o el botón cancelar de la Action Bar;<br>o el botón confirmar de la Action Bar dejando el campo nombre vacío. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil. |

Tabla 10. PFlu-07

| Identificador | PFlu-08 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Selección de Equipos* a la interfaz de *Selección de Porteros*. Modo táctil. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Selección de Equipos*.<br>Seleccionar en los spinner los nombres de los equipos que participaran en el partido.<br>Pulsar el botón de "Empezar Partido". |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, de interfaz *Selección de Equipos* a la interfaz de *Selección de Porteros*. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde la interfaz de *Selección de Equipos*.<br>Utilizar el comando "Local/Visitante".<br>Indicar el nombre del primer equipo.<br>Cuando se solicite indicar el nombre del segundo equipo.<br>Confirmar inicio de partido mediante comando de voz afirmativo. |
| **Resultado de la variante 1** | OK |
| **Variante 2 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Selección de Porteros* a la interfaz *Selección de Equipos*. |
| **Procedimiento de la variante 2** | Desde la interfaz de *Selección de Porteros.*<br>Utilizar el botón el botón *atrás* de Android. |
| **Resultado de la variante 2** | OK |
| **Gestión de errores posibles** | Error 1: Comando de voz inexistente<br>Error 2: Frase del usuario no reconocida<br>Error 3: Más de 3 fracasos en reconocimiento de voz<br>Error 4: Equipos existentes en la BBDD<br>Error 5: Nombre de ambos equipos no puede ser el mismo. |
| **Resultado de gestión de errores** | RespErr1: Se vuelve a pedir al usuario comando de voz.<br>RespErr2: Se le da al usuario varias frases posibles que pueden ser la que ha descrito.<br>RepErr3: Se le sugiere al usuario introducción mediante modo táctil.<br>RepErr4: Si el equipo que ha indicado el usuario por voz no existe en la BBDD se le sugiere que indique un nuevo valor para el nombre del equipo.<br>RepErr5: Se muestra mensaje de error indicando que es necesario seleccionar 2 equipos que no sean iguales. |

Tabla 11. PFlu-08

| Identificador | PFlu-09 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Selección de Porteros* a la interfaz de *Pantalla de Partido*. Modo táctil. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Selección de Porteros*.<br>Seleccionar en los spinner los nombres de los porteros que participaran como titulares en cada equipo.<br>Pulsar el botón de "Guardar Cambios". |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz de *Pantalla de Partido* a la interfaz *Selección de Porteros*. |
| **Procedimiento de la variante 1** | Desde la interfaz de *Pantalla de Partido*.<br>Utilizar el botón el botón *atrás* de Android;<br>o pulsar el botón home de la ActionBar y posteriormente la opción *Porteros*. |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Error 1: No exista ningún portero para al menos uno de los dos equipos. |
| **Resultado de gestión de errores** | RespErr1: Se le solicita al usuario que vuelva a las opciones de Edición de Equipos y que inserte al menos un jugador con la característica de Portero para ese equipo. |

Tabla 20. PFlu-09

| Identificador | PFlu-10 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Pantalla de Partido* a la interfaz de *Acciones*. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Pantalla de Partido*.<br>Pulsar el botón home de la ActionBar y posteriormente la opción *Acciones*. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz *Acciones* a la interfaz de *Pantalla de Partido*. |
| **Procedimiento de la variante 1** | Desde la pantalla de *Pantalla de Partido*.<br>Pulsar el botón home de la ActionBar y posteriormente la opción *Pantalla de Partido*. |
| **Resultado de la variante 1** | OK |

Tabla 21. PFlu-10

| Identificador | PFlu-11 |
|---|---|
| **Descripción línea principal** | Prueba del flujo de la interfaz, de interfaz *Pantalla de Partido* a la interfaz de *Estadísticas de Equipo <Local/Visitante>*. |
| **Procedimiento de la línea principal** | Desde la pantalla de *Pantalla de Partido*.<br>Pulsar el botón home de la ActionBar y posteriormente la opción *Estadísticas equipo <Local/Visitante>*. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba del flujo de la interfaz, volver de interfaz *Estadísticas equipo <Local/Visitante>* a la interfaz de |

| | |
|---|---|
| | *Pantalla de Partido*. |
| **Procedimiento de la variante 1** | Desde la pantalla de *Pantalla de Partido*.<br>Pulsar el botón home de la ActionBar y posteriormente la opción *Pantalla de Partido*. |
| **Resultado de la variante 1** | OK |

Tabla 12. PFlu-11

## 4.3.2 Pruebas de funcionalidad

| Identificador | PFun-01 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Insertar un Nuevo Equipo en la aplicación. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Editar Equipos;*<br>pulsar en el botón de la Action Bar *Añadir Equipo;*<br>escribir el nuevo nombre del equipo;<br>pulsar el botón de la Action Bar de *Confirmar.* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Insertar un Nuevo Equipo en la aplicación. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>*indicar el comando de voz para entrar en la interfaz de Editar Equipos;*<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Editar Equipo;*<br>indicar el comando de voz para añadir equipo;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Inserción;*<br>indicar el nuevo nombre del equipo y confirmarlo mediante comando de voz. |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Error 1: Equipo a insertar existente en la BBDD de la aplicación. |
| **Resultado de gestión de errores** | RespErr1: Se pide al usuario que indique un nombre para el equipo que no exista ya en la BBDD de la aplicación. |

Tabla 13. PFun-01

| Identificador | PFun-02 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Eliminar un Equipo de la aplicación. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Editar Equipos;*<br>pulsar en el nombre del equipo a eliminar*;*<br>pulsar el botón de la Action Bar de *Eliminar;*<br>confirmar la acción. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Eliminar un Equipo de la aplicación. Comandos de Voz. |

| | |
|---|---|
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Menú Principal*<br>*indicar el comando de voz para entrar en la interfaz de Editar Equipos;*<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Editar Equipo;*<br>indicar el comando de voz para editar equipo e indicar la posición en la lista del equipo a eliminar;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Jugadores;*<br>indicar que la acción deseada es la de eliminar el equipo y confirmar la acción mediante comando de voz. |
| **Resultado de la variante 1** | OK |

Tabla 14. PFun-02

| Identificador | PFun-03 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Insertar un jugador en la aplicación. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Editar Equipos;*<br>pulsar en el nombre del equipo en el cual juega el jugador que vamos a insertar*;*<br>pulsar el botón de la Action Bar para *Añadir Jugador;*<br>escribir los parámetros del jugador;<br>pulsar el botón de la Action Bar para *Confirmar.* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Insertar un jugador en la aplicación. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Menú Principal*<br>*indicar el comando de voz para entrar en la interfaz de Editar Equipos;*<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Editar Equipo;*<br>indicar el comando de voz para editar equipo e indicar la posición en la lista del equipo en el cual juega el jugador que vamos a insertar;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Jugadores;*<br>indicar el comando de voz para añadir jugador;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *EditarJugador;*<br>indicar uno a uno mediante voz los datos del jugador que vamos a añadir. |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Error 1: Jugador a Insertar ya existente en la BBDD de la aplicación.<br>Error 2: Falta alguno de los parámetros obligatorios |

| | |
|---|---|
| | para insertar al jugador. |
| **Resultado de gestión de errores** | RespErr1: Se pide al usuario que introduzca un nombre de jugador que no exista actualmente en la BBDD.<br>RespErr2: Se le pide al usuario que complete todos los datos necesarios para la inserción. |

Tabla 15. PFun-03

| Identificador | PFun-04 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Editar un jugador existente en la aplicación. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Editar Equipos;*<br>pulsar en el nombre del equipo en el cual juega el jugador que vamos a insertar*;*<br>pulsar el nombre del jugador que deseamos editar;<br>modificar los parámetros del jugador;<br>pulsar el botón de la Action Bar para *Confirmar.* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Insertar un jugador en la aplicación. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Menú Principal*<br>indicar el comando de voz para entrar en la interfaz de *Editar Equipos;*<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Editar Equipo;*<br>indicar el comando de voz para editar equipo e indicar la posición en la lista del equipo en el cual juega el jugador que vamos a insertar;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Jugadores;*<br>indicar el comando de voz para editar jugador e indicar el dorsal del jugador que deseamos editar;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *EditarJugador;*<br>indicar el parámetro/s a modificar y el valor de los mismos mediante comandos de voz y confirmar. |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Error 1: Falta alguno de los parámetros existentes tras la edición del jugador. |
| **Resultado de gestión de errores** | RespErr1: Se le pide al usuario que complete todos los datos necesarios para completar la edición del jugador. |

Tabla 16. PFun-04

| Identificador | PFun-05 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Eliminar un jugador de la aplicación. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Editar Equipos;*<br>pulsar en el nombre del equipo en el cual juega el jugador que vamos a insertar*;*<br>pulsar el nombre del jugador que se desea eliminar;<br>pulsar el botón *Eliminar* de la Action Bar de la interfaz *EditarJugador.*<br>Confirmar la eliminación del jugador. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Eliminar un jugador de la aplicación. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Menú Principal*<br>*indicar el comando de voz para entrar en la interfaz de Editar Equipos;*<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Editar Equipo;*<br>indicar el comando de voz para editar equipo e indicar la posición en la lista del equipo en el cual juega el jugador que vamos a eliminar;<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Jugadores;*<br>indicar el comando de voz para eliminar jugador;<br>indicar el dorsal del jugador a eliminar y confirmar la eliminación mediante voz. |
| **Resultado de la variante 1** | OK |

Tabla 17. PFun-05

| Identificador | PFun-06 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Comenzar un Nuevo Partido. Modo táctil. |
| **Procedimiento de la línea principal** | Desde el *Menú Principal*.<br>Pulsar el botón *Nuevo Partido;*<br>Seleccionar los equipos participantes en el partido en las listas desplegables y pulsar el botón *Empezar Partido;*<br>seleccionar los porteros titulares de cada equipo a través de las listas desplegables;<br>pulsar el botón *Guardar Cambios.* |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Comenzar un Nuevo Partido. Comandos de Voz. |
| **Procedimiento de la variante 1** | Desde el *Menú Principal*.<br>pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Menú Principal* e indicar el comando de voz para entrar en la interfaz de *Selección de Equipos;* |

| | |
|---|---|
| | pulsar en el botón de la Action Bar *Reconocimiento de Voz* de la interfaz *Selección de Equipos;*<br>indicar el nombre del primer equipo, posteriormente el nombre del rival y confirmar mediante comando de voz;<br>seleccionar los porteros titulares de cada equipo a través de las listas desplegables;<br>pulsar el botón *Guardar Cambios.* |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Error 1: Se selecciona el mismo equipo como equipo local y visitante.<br>Error 2: No existe al menos un portero en uno de los equipos al intentar comenzar un partido. |
| **Resultado de gestión de errores** | RespErr1: Se le pide al usuario que seleccione dos equipos que tengan diferente nombre para poder comenzar el partido.<br>RespErr2: Se le pide al usuario que inserte al menos un portero para los dos equipos seleccionados antes de comenzar el partido. |

Tabla 18. PFun-06

| Identificador | PFun-07 |
|---|---|
| **Descripción línea principal** | Prueba de funcionalidad. Introducción de una Acción de un Partido mediante comandos táctiles. |
| **Procedimiento de la línea principal** | Desde la Pantalla de Partido.<br>Pulsar el botón Comenzar para iniciar el partido actual;<br>seleccionamos la acción que se ha realizado durante el partido;<br>seleccionamos el jugador que ha realizado la acción. |
| **Resultado de la línea principal** | OK |
| **Variante 1 de la prueba** | Prueba de funcionalidad. Introducción de una Acción de un Partido mediante comandos de voz. |
| **Procedimiento de la variante 1** | Desde la Pantalla de Partido.<br>Indicar el comando de voz “Comenzar” para iniciar el partido actual;<br>Indicar mediante comando de voz la acción que se ha realizado durante el partido;<br>Indicar mediante comando de voz el equipo al que pertenece el jugador (Local/Visitante) y posteriormente el número o dorsal de dicho jugador. |
| **Resultado de la variante 1** | OK |
| **Gestión de errores posibles** | Err1: El usuario indica una acción que no está entre las posibles.<br>Err2: El usuario indica un jugador para realizar la acción que no existe en la BBDD. |
| **Resultado de gestión de errores** | RespErr1: No se realizará ninguna acción permitiendo al usuario otra entrada rápida de acción.<br>RespErr2: Se le pide al usuario que indique un nuevo jugador. |

Tabla 19. PFun-07

# Capítulo 5. Evaluación del sistema

En este capítulo se describe el método utilizado para llevar a cabo una evaluación del sistema por parte de personas externas al desarrollo de la aplicación. Para ello se ha utilizado el portal gratuito *Survio* que permite crear encuestas y registra los datos de los usuarios aportando estadísticas de las respuestas y análisis de datos.

## 5.1 Metodología de evaluación subjetiva

Se ha buscado sobre todo la calidad de la evaluación del sistema. Para ello se le ha entregado la aplicación a potenciales usuarios de la misma que por su relación con el mundo del balonmano podrían ser perfectamente usuarios finales en caso de ser publicada la aplicación.

## 5.2 Formato del cuestionario

El cuestionario a realizar por los usuarios para este fin consta de un total de 10 preguntas con múltiples respuestas. Es un cuestionario fácil de rellenar que no lleva un tiempo superior a los 5 min. A continuación se mostrará el formato de dicho cuestionario:

# Aplicación de Voz para Balonmano

Estimado Sr. / Sra.,

gracias por su visita. Rellenando esta breve encuesta, nos ayudará a obtener los mejores resultados para la evaluación de mi app de Android para la generación de estadísticas de Balonmano a través de comandos de voz.

**Indique que tipo de usuario es en su opinión respecto a las nuevas tecnologías** *

- Muy Avanzado
- Avandazo
- Medio
- Perfil Bajo
- Perfil muy Bajo

**¿Esta acostumbrado a utilizar aplicaciones a través de comandos de voz?** *

- Bastante
- Mucho
- Poco
- Nada

**Indique de 1 a 5 como cree que ha entendido del sistema sus comandos de voz** *

- 1
- 2
- 3
- 4
- 5

**Indique de 1 a 5 el nivel de entendimiento de las frases generadas por el sistema** *

- 1
- 2
- 3
- 4
- 5

**Indique de 1 a 5 la velocidad de interacción general con el sistema** *

- 1
- 2
- 3
- 4
- 5

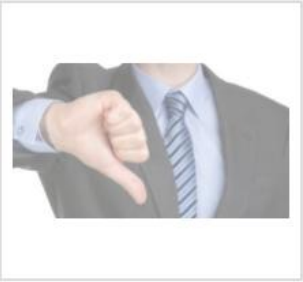
**Establezca de 1 a 5 la dificultad que le ha supuesto aprender los comandos de voz para intearactuar con el sistema.** *

- 1
- 2
- 3
- 4
- 5

**Sustituiría el sistema tradicional en papel por un sistema mediante dispositivo movil como el que se esta evaluando** *

- SI
- NO

**Considera util la posibilidad de utilizar comandos de voz en el escenario que los utiliza esta aplicación** *



SI NO

**A la hora de utilizar la aplicación. ¿Que preferencia tiene para la entrada de datos?** *

Comandos de Voz Comando Táctil

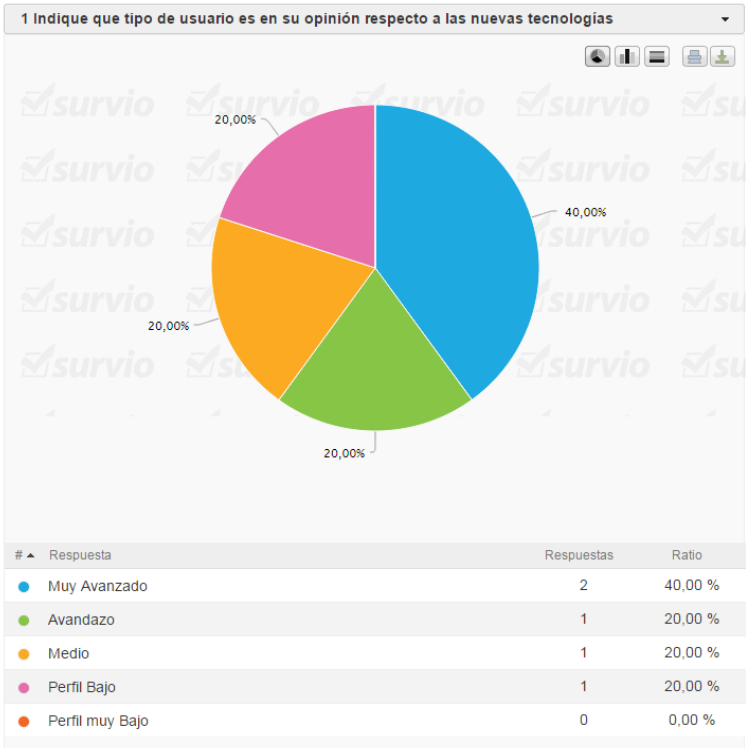
**Puntue de 1 a 10 lo util que encuentra la aplicación** *

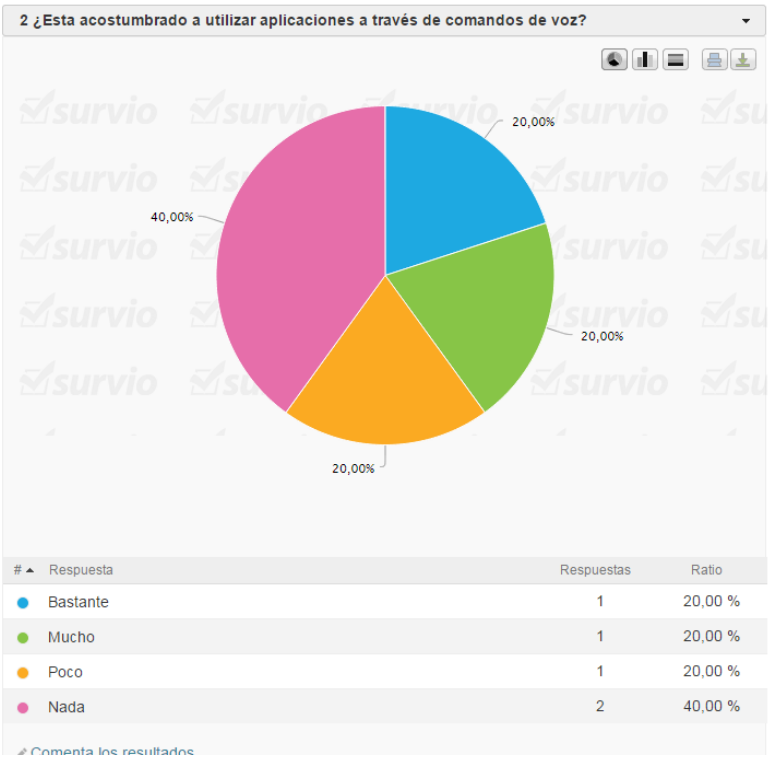
0 / 10

Se puede acceder a la página de la encuesta desde cualquier dispositivo que permita el acceso a Internet, y para ser completada, el usuario únicamente tiene que acceder al siguiente enlace Web:

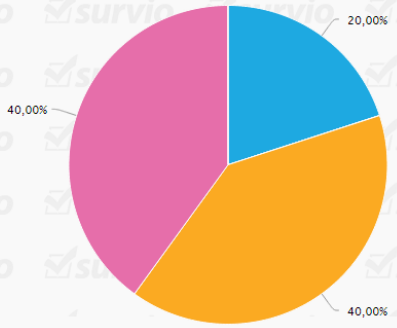
**http://www.survio.com/survey/d/R9B5C1N7L9O5C5U0A**

# 5.3 Resultados del cuestionario



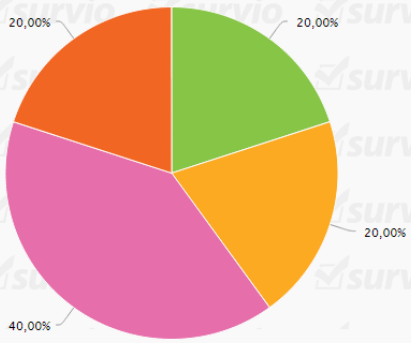
**3 Indique de 1 a 5 como cree que ha entendido del sistema sus comandos de voz**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| 1 | 1 | 1 | 20,00 % |
| 2 | 2 | 0 | 0,00 % |
| 3 | 3 | 2 | 40,00 % |
| 4 | 4 | 2 | 40,00 % |
| 5 | 5 | 0 | 0,00 % |

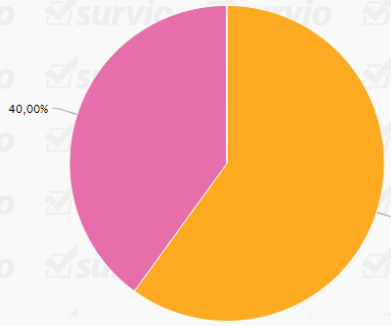
Comenta los resultados

**4 Indique de 1 a 5 el nivel de entendimiento de las frases generadas por el sistema**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| 1 | 1 | 0 | 0,00 % |
| 2 | 2 | 1 | 20,00 % |
| 3 | 3 | 1 | 20,00 % |
| 4 | 4 | 2 | 40,00 % |
| 5 | 5 | 1 | 20,00 % |

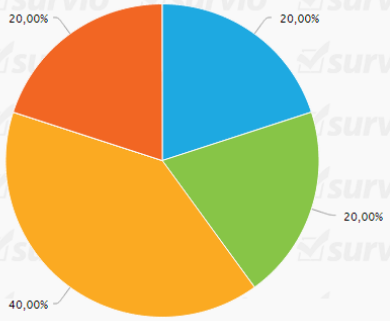
**5 Indique de 1 a 5 la velocidad de interacción general con el sistema**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| 1 | | 0 | 0,00 % |
| 2 | | 0 | 0,00 % |
| 3 | | 3 | 60,00 % |
| 4 | | 2 | 40,00 % |
| 5 | | 0 | 0,00 % |

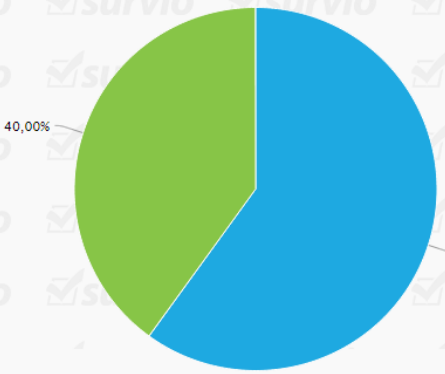
Comenta los resultados

**6 Establezca de 1 a 5 la dificultad que le ha supuesto aprender los comandos de voz para intearactuar con el sistema.**



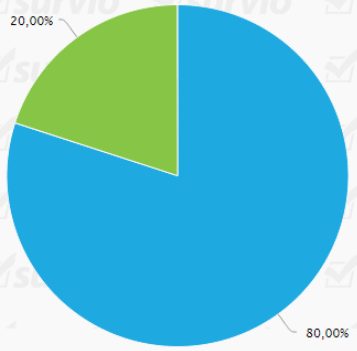

| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| 1 | | 1 | 20,00 % |
| 2 | | 1 | 20,00 % |
| 3 | | 2 | 40,00 % |
| 4 | | 0 | 0,00 % |
| 5 | | 1 | 20,00 % |

**7 Sustituiría el sistema tradicional en papel por un sistema mediante dispositivo movil como el que se esta evaluando**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| ● | SI | 3 | 60,00 % |
| ● | NO | 2 | 40,00 % |

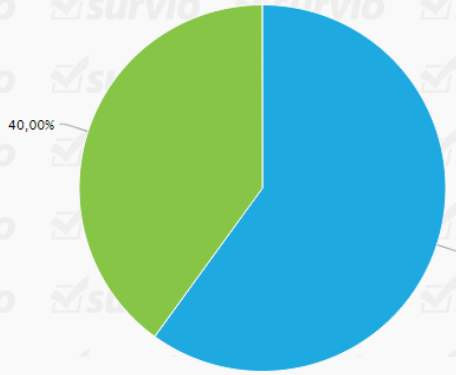
**8 Considera util la posibilidad de utilizar comandos de voz en el escenario que los utiliza esta aplicación**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| ● | SI | 4 | 80,00 % |
| ● | NO | 1 | 20,00 % |

Comenta los resultados

**9 A la hora de utilizar la aplicación. ¿Que preferencia tiene para la entrada de datos?**




| # | Respuesta | Respuestas | Ratio |
|---|---|---|---|
| ● | Comandos de Voz | 3 | 60,00 % |
| ● | Comando Táctil | 2 | 40,00 % |

Comenta los resultados

**10 Puntue de 1 a 10 lo util que encuentra la aplicación**


| Respuesta | Respuestas | Ratio |
|---|---|---|
| 1 / 10 | 0 | 0,00 % |
| 10 / 10 | 0 | 0,00 % |
| 2 / 10 | 0 | 0,00 % |
| 3 / 10 | 1 | 20,00 % |
| 4 / 10 | 0 | 0,00 % |
| 5 / 10 | 0 | 0,00 % |
| 6 / 10 | 0 | 0,00 % |
| 7 / 10 | 2 | 40,00 % |
| 8 / 10 | 1 | 20,00 % |
| 9 / 10 | 1 | 20,00 % |

Comenta los resultados

# 5.4 Conclusiones

- El cuestionario ha sido contestado por usuarios que tienen un grado de conocimiento sobre las nuevas tecnologías variado, de esta forma comprende un espectro variado del tipo posible de usuario que puede acceder a la aplicación.

- Es mayoritario el número de usuarios que están poco o nada acostumbrados a utilizar aplicaciones que utilizan comandos de voz.

- En general los usuarios que contestaron la encuesta están satisfechos con la forma en que el sistema ha entendido sus frases y han entendido bien las respuestas que les ha dado la aplicación.

- Los usuarios que respondieron la encuesta opinan que la velocidad de interacción con el sistema es media-alta.

- Los comandos de voz necesarios para interactuar con el sistema de acuerdo a los usuarios que respondieron a la encuesta son fáciles.

- Son mayoría los usuarios que opinan que el sistema de entrada de datos mediante comandos de voz es útil para el escenario en que se podrá utilizar esta aplicación y lo sustituirían por el método tradicional en papel.

- La evaluación total de los usuarios para el sistema es de un 6,8 sobre 10. Teniendo en cuenta que la media baja bastante por la calificación de un usuario que suspende la aplicación con una valoración total de 3 puntos sobre 10. Se puede concluir que la evaluación de la aplicación por los usuarios es bastante satisfactoria.

# 5.5 Metodología de evaluación objetiva

Se utilizará otro método de evaluación de la aplicación, en este caso se trata de un método más objetivo ya que depende de los resultados del usuario a la hora de utilizar la aplicación. La aplicación irá registrando una serie de datos estadísticos acerca de las intervenciones del usuario mediante comandos de voz y estos resultados nos permitirán evaluar el rendimiento de la aplicación.

Los datos estadísticos a guardar durante el uso de la aplicación cuando se realiza a través de comandos de voz serán:

- **Tiempo Total de Reconocimiento:** Tiempo en milisegundos utilizados para realizar el reconocimiento de voz de las frases que ha dicho el usuario durante la utilización de la aplicación. Este tiempo total luego se utilizará para obtener el tiempo medio de reconocimiento con el que podremos establecer el tiempo que utiliza de media la

aplicación para cada reconocimiento y saber si el tiempo de reconocimiento es adecuado.

- **Tiempo Total de Análisis:** Es el tiempo en milisegundos utilizado para analizar las frases que ha dicho el usuario y responder en consecuencia con las acciones solicitadas por el mismo. Este tiempo total luego se utilizará para obtener el tiempo medio de análisis y conocer si el tiempo de actuación de la aplicación es el adecuado.

- **Número Total de Reconocimientos:** Es el número total de reconocimientos realizado por la aplicación, tanto acertados como fallidos. Nos permitirá realizar los cálculos de tiempos medios tanto de reconocimiento como de análisis.

- **Numero de Reconocimientos Fallidos:** Número total de reconocimientos en los que la aplicación no ha sido capaz de realizar un reconocimiento o análisis satisfactorio de las peticiones realizadas por el usuario. Comparando este dato con el número total de reconocimientos es una medida de evaluación de la respuesta de la aplicación.

Para poder consultar estas estadísticas de la aplicación se habilita una interfaz desde la que se puede acceder desde el Menú Principal de la aplicación, llamada "Resultados de Voz", la cual esta implementada a través de la clase ResVoz.class y que tiene la siguiente interfaz:

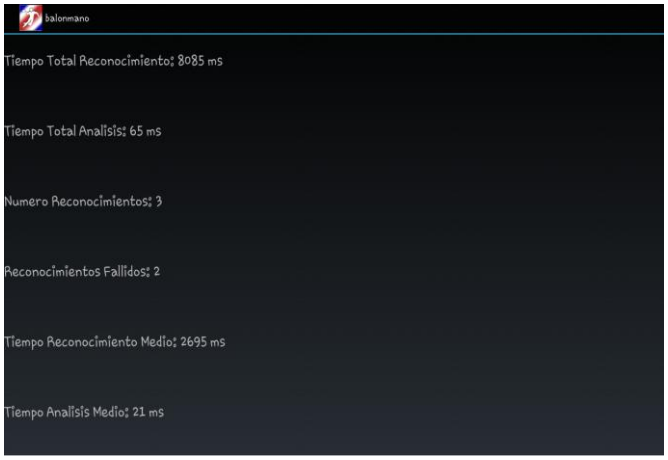



Figura 111. Interfaz de Resultados de Voz

# Capítulo 6. Presupuesto

Este capítulo servirá para mostrar el presupuesto del Proyecto Final de Carrera. Este será el presupuesto final que lleva desarrollar la idea final planteada en esta memoria para poder ser posteriormente comercializada si fuera necesario y se quisiera en el mercado de las aplicaciones móviles.

Esto contempla realizar un desglose de la duración de las distintas fases y tareas, evaluar los costes de personal, materiales y recursos totales que se han utilizado en este Proyecto Final de Carrera.

## 6.1 Evaluación de la duración de las tareas

- **Fase de estudio**
  - Estudio de los Sistemas de diálogo = **15 días**
  - Estudio del Sistema operativo Android
    - Número de desarrollo actual: 4.4 KitKat = **10 días**
    - IDE para el desarrollo: Eclipse = **5 días**
    - Lenguaje de programación utilizado para desarrollo: Java = **30 días**

- **Fase de planificación**
  - Planificación de la estructura de la Bases de Datos = **2 días**
  - Planificación y formato de la Interfaz gráfica = **15 días**
  - Planificación del funcionamiento del Reconocimiento de voz en Android = **4 días**
  - Diseño de los Comandos de voz a utilizar en la aplicación = **6 días**

- **Fase de desarrollo**
  - Desarrollo de la interfaz gráfica = **30 días**
  - Desarrollo de la comunicación con BBDD = **7 días**
  - Desarrollo del módulo de Análisis de Voz = **24 días**
  - Desarrollo del módulo de Síntesis de Voz = **16 días**

- **Fase de pruebas**

- Pruebas de flujo de la interfaz gráfica = **3 días**
- Pruebas funcionales de la aplicación = **13 días**

- **Fase de evaluación**
  - Recogida de datos por parte de usuarios finales = **3 días.**
  - Estudio de los datos recogidos y obtención de conclusiones de los mismos = **5 días.**

- **Fase de documentación**
  - Memoria del Proyecto Final de Carrera **= 30 días**
  - Preparación de la presentación **= 7 días**

Teniendo en cuenta que se han distribuido las jornadas en una duración de 6,5 horas por cada día de trabajo. Finalmente cada una de las fases del proyecto tiene la siguiente duración.

| Fase | Duración en Días | Duración en horas |
|---|---|---|
| **Estudio** | 60 días | 390 h |
| **Planificación** | 27 días | 175.5 h |
| **Desarrollo** | 77 días | 500.5 h |
| **Pruebas** | 16 días | 104 h |
| **Evaluación** | 8 días | 52 h |
| **Documentación** | 37 días | 240.5 h |
| **TOTAL** | 225 días | 1462.5 h |

# 6.2 Recursos utilizados

Se pretende especificar todos los recursos necesarios empleados en la elaboración de este proyecto. Para ello es necesario realizar un cálculo de costes efectivo dado que será el presupuesto a aprobar por el cliente para implementar la solución diseñada.

Se procede por tanto a especificar los recursos necesarios, tanto materiales como humanos. En primer lugar se exponen los recursos materiales de los que se requiere para la implementación del proyecto:

| RECURSOS HARDWARE NECESARIOS | | | | | |
|---|---|---|---|---|---|
| **DESCRIPCIÓN** | **UNIDADES** | **COSTE UNIDAD** | **VIDA ÚTIL ESTIMADA** | **AMORTIZACIÓN** | **COSTE TOTAL** |
| Dispositivo Tablet Samsung Galaxy Tab 3 10.1 | 1 ud | 299,00 € | 24 meses/ud | 12,45 €/mes | 93.40 € |
| Portatil HO Oavilion 15-n258ss de 15.6'' | 1ud | 489,00 € | 36 meses/ud | 13,60€/mes | 102.00 € |
| **TOTAL** | | | | | **195.40 €** |

Finalmente se exponen los costes del software necesario empleado:

| RECURSOS MATERIALES NECESARIOS | | | |
|---|---|---|---|
| **DESCRIPCIÓN** | **UNIDADES** | **COSTE UNIDAD** | **COSTE TOTAL** |
| Windows 7 Proffesional Edition | 1 ud | INC. EN HP | 0,00€ |
| Microsof Officce 2007 | 1ud | 219,00 € | 219,00 € |
| SDK de Android | 1ud | GRATIS | 0,00€ |
| Eclipse Framework | 1ud | GRATIS | 0,00€ |
| Administrador de la BBDD SQLite | 1ud | GRATIS | 0,00€ |
| **TOTAL** | | | **219.00 €** |

Finalmente daré una estimación de los costes de los recursos de personal que se utilizaron durante el proyecto. Para ello tendremos en cuenta la categoría profesional de las personas que han intervenido en el mismo y por tanto la diferencia de salario entre las mismas de la siguiente forma:

- Jefe de Proyecto: Ingeniero Senior = A razón de 40 €/h
- Programador: Ingeniero Junior = A razón de 25 €/h

| RECURSOS PERSONALES DE LOS QUE SE DISPONE | |
|---|---|
| NOMBRE Y DEDICACIÓN | **COSTE** |
| **David Griol – Jefe de Proyecto**<br><br>**Horas dedicadas al proyecto**<br>**300 Horas * 40€/h** | 12.000,00 € |
| **Jonathan Caballero – Programador**<br><br>**Horas dedicadas al proyecto**<br>**1462.5 Horas * 25€/h** | 36.562,50 € |
| **TOTAL** | 48.562,50 € |

# 6.3 Presupuesto final

Finalmente teniendo en cuenta todos los recursos necesarios para llevar a cabo este proyecto, el resultado del presupuesto final sería:

| PRESUPUESTO FINAL | |
|---|---|
| TIPO DE RECURSO | **COSTE** |
| **Recursos Hardware** | 195,40 € |
| **Recursos Materiales** | 219,00 € |
| **Recursos Personales** | 48.562,50 € |
| **TOTAL** | **48.976,90 €** |

Por tanto, el presupuesto estimado para llevar a cabo este proyecto asciende a la cantidad de CUARENTA Y OCHO MIL NOVECIENTOS SETENTA Y SIETE €UROS.

# Capítulo 7. Conclusiones y trabajos futuros

Este capítulo servirá para documentar las conclusiones obtenidas de la realización de este Proyecto Final de Carrera: revisar los resultados obtenidos, identificar si los objetivos fijados inicialmente se han cumplido y si la evaluación por parte de los usuarios es satisfactoria.

Teniendo en cuenta que el mundo de las aplicaciones móviles supone actualizaciones continuas, en este capítulo se indicarán algunas de las líneas a desarrollar para mejorar en el futuro las funcionalidades de la aplicación así como su eficiencia.

## 7.1 Conclusiones

Una vez que el Proyecto Fin de Carrera ha sido concluido pasaré a determinar si los objetivos fijados inicialmente han sido cumplidos y si los resultados obtenidos han sido satisfactorios.

En este proyecto se ha desarrollado una aplicación para sistemas móviles que trabajan con el Sistema Operativo Android. Se trata de una aplicación cuya finalidad es realizar las estadísticas en tiempo real de un partido de balonmano para que puedan ser analizadas por el cuerpo técnico de un equipo.

Las funcionalidades que ofrece la aplicación se reparten en diferentes menús principales por los que puede navegar el usuario mediante comandos táctiles o sencillos comandos de voz. Estos menús están interconectados entre sí a través de la BBDD de la aplicación donde se guarda toda la información disponible para el usuario.

La **funcionalidad de Resultados** permite al usuario consultar el tanteo final de aquellos partidos que se han realizado las estadísticas a través de la aplicación. La pantalla de resultados permite filtrar por equipo de forma que se vean de un rápido vistazo los resultados concretos de un equipo en particular.

La **funcionalidad de Estadísticas** permite al usuario consultar datos estadísticos resultantes de los partidos que se han introducido a través de esta aplicación. Serán datos estadísticos generales de toda la temporada y que se clasifican por equipos, diferenciando los datos de cada jugador de forma individual dentro del equipo. Además de los datos habituales introducidos por el

usuario, se mostrarán porcentajes de eficacia que permitan identificar rápidamente los jugadores más determinantes.

La **funcionalidad de Gestión de Equipos** dará facilidad al usuario para gestionar los diferentes equipos de las ligas que maneje, pudiendo editarlos en todo momento para incluir nuevos jugadores, eliminarlos, añadir nuevos equipos a la liga y la gestión de sus datos personales.

La **funcionalidad de Partido** es la línea principal de la aplicación y la funcionalidad más importante en la que el usuario puede realizar en tiempo real la introducción de las diferentes acciones que transcurren durante un partido. La posibilidad de la introducción de datos mediante voz, además del tradicional método táctil enriquece las posibilidades para el usuario, ya que le permite poder estar pendiente en todo momento del partido sin perderle la vista, lo que pudiera ocasionar que se perdiera algunas de las acciones que transcurren.

La plataforma que me ha permitido desarrollar los componentes de esta aplicación, sobre todo en lo que se refiere al reconocimiento y síntesis de voz ha sido el sistema operativo para sistemas móviles Android. Este sistema operativo destaca por ser el sistema operativo más utilizado en el mundo para este tipo de dispositivos. Además su potente API para desarrollo de aplicaciones facilita enormemente al desarrollador la gestión de funcionalidades relacionadas con la voz.

Entre los objetivos planteados inicialmente de forma principal estaba el desarrollo de una aplicación para dispositivos móviles con sistema operativo Android que permitiera la interacción hombre-máquina a través de la voz utilizando lenguaje natural. Este objetivo se ha cumplido con la aplicación realizada.

Se han encontrado diferentes dificultades durante la planificación de la aplicación que se han tenido que afrontar durante el proceso la fase de desarrollo:

- Se estima inicialmente que el método ideal de reconocimiento de voz es tener un hilo en modo de “siempre a la escucha” que permita al usuario emitir comandos de voz de forma continua. El problema de utilizar este hilo es que al estar el proceso continuamente a la escucha podría ralentizar el resto de la aplicación así como el consumo continuo de batería que es un elemento escaso y crítico en los dispositivos móviles.
  El siguiente método que se pensó es utilizar un sistema de “touch and talk”, en el que el usuario pulsara un botón físico del dispositivo cada vez que quisiera realizar un reconocimiento de voz. El problema que surgía de utilizar este método era que los botones físicos del dispositivo es que a la larga no están preparados para tantas pulsaciones y se pueden terminar estropeando.

De forma que finalmente el método utilizado es el de un sistema “touch and talk”, pero en este caso se ha utilizado un botón virtual que esta embebido en la interfaz de la aplicación. Se ha utilizado esta solución puesto que la pantalla de la aplicación es la parte del dispositivo móvil más preparada para pulsaciones sucesivas.

- Otra de las dificultades que surgieron es que para utilizar el reconocimiento de voz de Google es necesario que el dispositivo esté conectado a Internet. Este problema se soluciona por parte de Google para Sistemas Operativos superiores a la versión 4.2 de Android. Sin embargo para dispositivos anteriores la posible solución sería que el usuario instalara una aplicación externa para el reconocimiento de voz.

Además la motivación era realizar una aplicación que aunque se pudiera encontrar algo similar en el mercado supusiera una diferenciación de lo demás existente en el mercado. Esto se ha realizado mezclando el objetivo de la aplicación con el mundo del deporte. Se ha analizado una necesidad real, surgida de los delegados de equipo de un deporte colectivo como es el balonmano, que a la hora de realizar las estadísticas en tiempo real, ya fuera por un medio tradicional, a través de papel, como con un medio multimedia, a través de la pantalla táctil de un dispositivo móvil, lo cierto es que cualquiera de ambos métodos hacían que al anotar una acción, los técnicos se perdieran parte de las acciones que transcurrían durante el partido, pues dicho deporte es bastante dinámico, y transcurren varias acciones muy seguidas. La utilización de los comandos de voz para introducir los datos en tiempo real suponen la diferenciación característica de esta aplicación.

Al analizar los resultados de la evaluación de la aplicación podemos concluir que si bien, en principio los potenciales usuarios son un poco reticentes a la utilización de la misma, sobre todo por el pequeño aprendizaje que supone por su parte del aprendizaje de los comandos de voz necesarios para su utilización, si que están dispuestos a intentar trabajar con ella, puesto que la idea de poder introducir datos sin perderse las acciones que transcurren les supone un avance en este sentido.

# 7.2 Trabajos futuros

Finalizado el Proyecto, se pasa a detallar una serie de líneas de desarrollo futuro que permitirán mejorar la aplicación desarrollada.

### 7.2.1 Funcionalidades Adicionales a las interfaces

**Interfaz de Pantalla de Partido**

Posibilidad de completar las acciones a realizar por los jugadores durante el partido. Por ejemplo añadir eventos como, el lugar desde el que ha realizado el lanzamiento, a que zona de la portería se ha realizado dicho lanzamiento, etc.

Esto además puede llevar a cabo la posibilidad de añadir “alarmas” en la aplicación ante ciertos eventos que se repiten con frecuencia, de forma que amplíen la información al usuario. Dichas alarmas pueden estar en la línea: el jugador número 7 del equipo contrario ha realizado los últimos 3 lanzamientos a la zona izquierda de la portería o el portero del equipo contrario nos ha parado 3 penaltis en su zona derecha y ha encajado todos los que hemos lanzado en su zona izquierda.

**Añadir Nuevos Deportes**

La aplicación actual está orientada al deporte del balonmano en particular, pero sería fácilmente extensible a cualquier otro deporte de equipo.

Para ello se podría comenzar por incluir las acciones propias de cada deporte a la interfaz de la pantalla principal de equipo. Además sería necesario incluir en alguna de las interfaces iniciales, por ejemplo la de selección de equipos participantes en el partido, una opción que sea seleccionar el deporte que van a practicar dichos equipos.

**Gestión de Equipos con más datos**

En las interfaces de gestión de datos de equipo, se podrían incrementar datos de los jugadores para complementar la información de los mismos, de forma que la aplicación además sirva como pequeña Base de Datos para la gestión de la información de equipos. Se podrían añadir datos tales como:

- Edad
- Altura
- Peso
- Fotografía
- Medidas Antropométricas.

Además esta cuestión puede llevar también a modificar la pantalla de estadísticas post-partido y en tiempo real para hacerlas mucho más completas, por ejemplo mostrando de forma individualizada una pantalla por jugador con los datos del mismo y las estadísticas partido a partido y las completas de la temporada.

**Gestión de diferentes torneos para un mismo equipo**

Actualmente la aplicación permite gestionar los datos y estadísticas de un equipo durante una temporada completa, pero la cuestión es, que durante una temporada un mismo equipo suele disputar diferentes competiciones y suele ser interesante diferenciar las estadísticas del equipo para cada una de estas diferentes competiciones.

En esta línea por tanto sería interesante añadir la posibilidad de gestión de torneos para un mismo equipo.

### 7.2.2 Idiomas

El mundo del deporte es de ámbito internacional. Y los más extendidos en concreto son los deportes de equipo, aunque en concreto el balonmano, se encuentra entre los más minoritarios, también esta muy extendido por todo el mundo. Esto implica que la aplicación podría ser utilizada en diversos campeonatos por usuarios de prácticamente cualquier parte del mundo, en cuanto a funcionalidad se requiere. La única barrera, sobre todo en la cuestión de entrada de información mediante voz, supone el cambio de idioma.

Una funcionalidad a añadir sería incluir una opción inicial en el menú principal que permita seleccionar el idioma del usuario, inicialmente se añadirían aquellos idiomas de los países donde está más extendido el deporte concreto del balonmano: francés y alemán principalmente.

El requisito fundamental para añadir este servicio se encuentra en los módulos de reconocimiento y síntesis de voz, de forma que sea posible identificar las instrucciones del usuario en estos idiomas. Las clases de reconocimiento y síntesis de voz en Android (RecognizerIntent y TextToSpeech), permiten realizar esta tarea.

### 7.2.3 Subida de datos en tiempo real a servidor externo para consulta por otros

Una funcionalidad interesante a añadir a la aplicación sería la posibilidad de subir en tiempo real la información de las acciones que introduce el usuario en la aplicación a un servidor externo. El objetivo sería que este servidor se trate de un servidor web capaz de suministrar una página web dinámica que se vaya actualizando a medida que transcurre el partido. Esto supone que una persona que no pueda ver el partido en directo sí que sea capaz de seguirlo de forma activa a través de una conexión al servidor web.

### 7.2.4 Entrada de información a través de video

En la línea de hacer la aplicación lo más multimodal posible, se podría ampliar las funcionalidades de la misma estudiando la posibilidad de introducir datos en la misma a través de video, de forma que a través de la cámara del dispositivo, mientras se graba en tiempo real un partido del deporte del cual se quieren realizar las estadísticas, la aplicación sea capaz de identificar el jugador que ha realizado la acción (por ejemplo con ayuda de herramientas de reconocimiento facial y corporal), el tipo de acción que ha realizado y completar la información de acciones de la base de datos únicamente con la entrada de video.

# GLOSARIO

**API:** *Application Programming Interface.* Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**TRIM:** En informática una orden TRIM permite al Sistema Operativo comunicarle a una unidad de estado sólido qué bloques de datos ya no están en uso, como los datos dejados al borrar ficheros.

**OpenGL:** Es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

**Hotspot:** En el contexto de las comunicaciones inalámbricas es un dispositivo o lugar que ofrece acceso a internet a través de una red inalámbrica y un enrutador conectado a un proveedor de servicios de Internet.

**Thetering:** Se denomina al proceso por el cual un dispositivo móvil con conexión a Internet actúa como pasarela para ofrecer acceso inalámbrico a la red a otros dispositivos, asumiendo dicho dispositivo móvil un papel similar al de un módem o enrutador inalámbrico.

**JIT:** *just-in-time* conocido como "compilación en tiempo de ejecución" en castellano. Se trata de una técnica para mejorar el rendimiento de sistemas de programación que compilan a bytecode, consistente en traducir el bytecode a código máquina nativo en tiempo de ejecución.

**JavaScript:** Lenguaje de programación usado para el desarrollo de aplicaciones cliente-servidor en páginas HTML.

**Chrome:** Navegador web desarrollado por Google.

**Bytecode:** Código intermedio entre el código fuente (escrito por el desarrollador en cualquier lenguaje de programación) y el código máquina (el que es interpretado realmente por un dispositivo, basado en código binario). Se le suele tratar como un fichero binario que contiene un programa ejecutable.

**NFC:** *Near Fiel Communication* o "Comunicación de Campo Cercano" en castellano. Tecnología de comunicación inalámbrica de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

**YAFFS:** Yet Another Flash File System o "Sólo otro sistema de ficheros flash" en castellano. Primer sistema de ficheros que fue diseñado específicamente para Memoria Flash NAND.

**Ext4:** Sistema de archivos transaccional basado en un sistema de archivos con registro diario y que está orientado principalmente a distribuciones Linux.

**Widget:** Pequeño programa que da acceso a funciones que se usan frecuentemente.

**MKV**: es un formato contenedor de estándar abierto, un archivo informático que puede contener una cantidad ilimitada de vídeo, audio, imagen o pistas de subtítulos dentro de un solo archivo.1 Su finalidad es la de servir como formato universal para el almacenamiento de contenidos audiovisuales y multimedia, como películas o programas de televisión, videojuegos, imágenes y textos.

**Wifi:** mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

**Bluetooth**: es una especificación industrial para Redes Inalámbricas que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda de los 2,4 GHz.

**Kernel**: Es la parte software del Sistema Operativo responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

GPS: es un sistema que permite determinar en todo el mundo la posición de un objeto con una precisión de unos pocos metros a través de una red de 24 satélites sincronizados para cubrir toda la superficie de la Tierra.

# BIBLIOGRAFÍA

**[LOP05]**

López-Cózar, R., and Araki, M.: „Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment‟ (Wiley, 2005).

**[LOP]**

Sistemas de diálogo hablado y multimodal. Disponible: <http://www.ugr.es/~rlopezc/sistemas_dialogo.htm> [10 de Septiembre de 2012]

**[LLI06]**

LLISTERRI, J. (2006) "Introducción a los sistemas de diálogo", in LLISTERRI, J.- MACHUCA, M. J. (Eds.) *Los sistemas de diálogo*. Bellaterra - Soria: Universitat Autònoma de Barcelona, Servei de Publicacions - Fundación Duques de Soria (Manuals de la Universitat Autònoma de Barcelona, Lingüística, 45). ISBN: 84-490-2459-5.

**[PAL]**

Muestra de voz de Hal. Disponible: <http://www.palantir.net/2001/sounds.html> [10 de Septiembre de 2012]

**[JOA]**

http://liceu.uab.es/~joaquim/speech_technology/tecnol_parla/dialogue/dialogo_general/sistemas_dialogo.html (Ultima visita 10/09/2012).

**[LAR03]**

Traum, D., Larsson, S., 2003. The Information State Approach to Dialogue Management. Current and New Directions in Discourse and Dialogue. Kluwer Academic Publisher.

**[DUT97]**

Dutoit, T. (1997). An introduction to text-to-speech synthesis. Dordrecht: Kluwer. (p. 17)

**[DEF]**

Definición de Android. Disponible: <http://conceptodefinicion.de> [10 de Septiembre de 2012]

**[ACID]**

Comparativa de buenos navegadores. Disponible: <acid3.acidtests.org/> [12 de Septiembre de 2012]

**[ANG]**

Ángel J. Vico, autor del blog La Columna 80. Post: “Arquitectura de Android”.

**[GAB11]**
Gabriel Ruiz Miranda: “Android y la discapacidad Visual”. (Septiembre 2011).

**[GRI07]**
Griol, David. Desarrollo y evaluación de Diferentes Metodologías para la Gestión Automática del Diálogo. Tesis Doctoral. Universidad Politécnica de Valencia, 2007.

**[WIK12]**
Ampliación de las características de los diferentes sistemas operativos de Android. Disponible:
<http://es.wikipedia.org/wiki/Android> [14 de Septiembre de 2012]

**[OHM12]**
Ampliación de las características de los diferentes sistemas operativos de Android. Disponible:
<http://www.ohmyphone.com/android/los-dulces-sistemas-operativos-de-android/> [14 de Septiembre de 2012]

**[MOV12]**
Ampliación de las características de los diferentes sistemas operativos de Android. Disponible:
<http://movileslibresbaratos.blogspot.com/2012/05/android-sus-diferentes-versiones.html> [14 de Septiembre de 2012]

**[PET12]**
Definición y tipos de IDE. Disponible:
<http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf> [14 de Septiembre de 2012]

**[JYC]**
John Freddy Vega y Christian Van Der Henst S. “El Presente del web: HTML 5”.

**[TAR12]**
Instalación de JDK de Java. Disponible:
<http://www.taringa.net/posts/ciencia-educacion/10979915/Instalar-JDK-y-eclipse-en-Windows-para-programar-en-Java.html> [14 de Septiembre de 2012]

**[ZAR12]**
Definición de JDK. Disponible:
<http://zarza.usal.es/~fgarcia/doc/tuto2/A_I.htm> [14 de Septiembre de 2012]

**[LUI12]**
Instalación de SDK de Java. Disponible:

<http://www.luiskano.net/blog/2010/04/28/tutorial-instalar-java-jdk-en-windows-7/> [14 de Septiembre de 2012]

**[PRO12]**
Definición de SDK. Disponible: <http://programaciondesarrollo.es/%C2%BFque-es-sdk/> [14 de Septiembre de 2012]

**[SGO12]**
Preparación de entorno de desarrollo Android en Windows. <http://www.sgoliver.net/blog/?p=1267> [14 de Septiembre de 2012]

**[SEB12]**
Ventajas y Desventajas de Android. Disponible: <http://sebaushares.blogspot.com.es/2011/12/android-es-un-sistema-operativo-basado.html> [14 de Septiembre de 2012]

**[TEA12]**
Comparación de ventajas y desventajas entre sistemas operativos móviles. Disponible: <http://team-soporte.es.tl/Android,-iOS-y-Windows-Phone-Ventajas-y-Desventajas.htm> [14 de Septiembre de 2012]

**[APR12]**
Definición de Malware. Disponible: <http://aprenderinternet.about.com/od/SeguridadPrivacidad/a/Que-Es-Malware.htm> [14 de Septiembre de 2012]

**[TUE12]**
Comparativa entre Sistemas operativos. Disponible: <http://www.tuexpertomovil.com/2011/11/03/comparativa-android-4-0-de-google-vs-ios-5-de-apple/> [14 de Septiembre de 2012]

**[MAE12]**
Definición de Widgets. Disponible: <http://www.maestrosdelweb.com/editorial/widgets/> [14 de Septiembre de 2012]

**[TAR12_02]**
Modelo de personalización de un dispositivo Android. Disponible: <http://www.taringa.net/comunidades/milestone/2089077/_Aporte_-_Personaliza-al-extremo-tu-android_.html> [14 de Septiembre de 2012]

**[XAT12]**
Comparativa de reconocedores de voz Android e iOS. Disponible: <http://www.xatakandroid.com/aplicaciones-android/siri-se-enfrenta-al-nuevo-google-search-y-parece-que-google-ha-vencido> [14 de Septiembre de 2012]

**[XAT14]**
Cuota de mercado de Android en función de los dispositivos: <http://www.xatakandroid.com/mercado/android-4-4-kitkat-ya-esta-en-el-13-6-de-los-dispositivos> [14 de Septiembre de 2014]

**[IAS12]**
Estudio Mobile Marketing. Disponible: <http://www.iabspain.net/wp-content/uploads/downloads/2013/09/V_Estudio_Mobile_Marketing_version_corta.pdf> [14 de Septiembre de 2012]

**[TIP14]**
Tipos de Dispositivos móviles. Disponible: <http://leo.ugr.es/J2ME/INTRO/intro_4.htm> [14 de Septiembre de 2012]

**[COL]**
CLODOALDO, R, S. "Programación en Android" ISBN eBook en PDF: 9788436954319.

**[COM]**
Comparativa con otros sistemas operativos. Disponible: <http://www.androidcurso.com/index.php/curso-android-basico/tutoriales-android-basico/31-unidad-1-vision-general-y-entorno-de-desarrollo/98-comparativa-con-otras-plataformas> [12 de Mayo de 2014]

**[NAZ]**
NAZIM, B. "Android 4: Principios del desarrollo de aplicaciones Java".

**[GIR]**
GIRONÉS, J.T. "El gran Libro de Android".

**[DEV14]**
*Get the Android SDK.* Disponible: <http://developer.android.com/sdk/index.html> [16 de Mayo de 2014].